Visual analysis of abstract multi-dimensional data with parallel coordinates.

Geng, Zhao

Visual Analysis of Abstract Multi-Dimensional Data with Parallel Coordinates

Zhao Geng

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Doctor of Philosophy

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

2012

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed (candidate)

Date 10/02/2013

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ... (candidate)

Date 10/02/2013

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ... (candidate)

Date 10/02/2013

Abstract

Parallel coordinates, introduced by Inselberg and Dimsdale [Ins09, ID90b], is a widely used visualization technique for exploring large, multi-dimensional data sets. It is powerful in revealing a wide range of data characteristics such as different data distributions and functional dependencies [KK96]. However, one of the limitations with parallel coordinates is the clutter problem caused by rendering more polylines than available pixels. Overlapped lines often obscure the underlying patterns of the data, especially in areas with high data density. In addition to large volume, data sets with high dimensionality could also bring difficulties for parallel coordinates to present in a limited screen space.

This thesis describes a literature study and a practical research in the field of information visualization, with special emphasis on how to overcome the clutter and overplotting in parallel coordinates when rendering large and high-dimensional data sets. Multi-dimensional data sets are generated from different domains, in our work, we mainly focus on two of them, namely scientific animal sensor data and natural language text data. The first data set suffers from large volume, but has relatively small number of dimensions. Whereas the second data set suffers from high dimensionality, but has fewer data elements. For the *Othello* data set we are not visualizing the original text itself. Instead, we utilize statistics to extract semantic features from the original document. These features are represented in numerical data format with high dimensionality.

To start with, we survey the variety of state-of-the-art, off-the-shelf information visualization tools in order to help researchers from Human Computer Interaction gain insight into their experimental data. Along this process, we have learned a broad overview of the information techniques with respect to their applicability. Next, we propose our own algorithms to overcome the overplotting in parallel coordinates introduced by visualizing large data sets. The first algorithm presented is called angular histograms. This technique is a frequency-based approach to large, high-dimensional data visualization. In our second proposed algorithm, we utilize the Markov Chain model to compute an n-dimensional joint probability for each data tuple based on a two-dimensional binning method. This probability value can be utilized to guide the user for selection and brushing in parallel coordinates. Later, we have developed two interactive visualization systems to explore the variations of *Othello* German translations. A Focus + Context parallel coordinates system is proposed for in-depth document term comparison and exploration. In addition, a coordinated multiple views visualization is further implemented to enable the document segment comparison.

Acknowledgements

I would not be able to finish this thesis without the support of a number of people who have had a direct or indirect impact on my three years of study for a PhD in Swansea. Having a retrospective view on this long journey, I'm very thankful to all these people. Especially I would like to thank my supervisor Dr. Robert S. Laramee for his endless and tireless guidance, support, care, patience and encouragement during my three years of study. I can still remember three years ago when I started my PhD, at the time I was a student lacking self-confidence and very confused about the future. Now my confidence has grown strongly, not only within my academic field, but also in my daily life. I'm no longer confused about my life and future, because I have found great interest and gratification in my research. This all attributes to my supervisor. I appreciate everything that I have learnt from him, especially over one hundred copies of minutes of meeting protocols, which have recorded every piece of painstaking work we have done together. These milestones can be accessed on `http://cs.swan.ac.uk/~cszg/minutes/`.

I am very thankful to the whole Department of Computer Science. I also want to give special thanks to the Visual and Interactive Computing group; in particular Jason Xie for his encouragement and advice when I found it difficult to finish my PhD and Dr. Rita Borgo for her valuable advice on my first job interview. I also would like to thank some research colleagues who have been a pleasure to work with: Tony Mcloughlin, Edward Grundy, Zhenmin Peng, Hui Fang, Dan Lipsa, Matthew Edmunds, Liam O'Reilly, David Chung, Ben Spencer and Zhang Haizhong. Special thanks go to James Walker for proof reading part of the thesis.

I also want to thank co-authors of our published research papers and other collaborators : Dr. Tom Cheesman, Dr. Jonathan Roberts, Dr. Rick Walker and Dr. Fernando Loizides. In addition, I would like to thank Thomson Reuters where I spent ten weeks working and having a wonderful summer in India. I also thank to the colleagues I have worked with at Thomson Reuters, which was truly a great pleasure.

My final words are dedicated to my family. Thank to my mother, Lu GuangJv, for her unconditional love and support. Without her constant encouragement by no means I can research to this point. My father, Geng JianGuo, passed away in the year 2010, which was the most devastating moment in my life. I even thought about withdrawing my PhD at the time. Two years have now passed, and I still miss him every single day. He has been the source of my momentum in overcoming so many obstacles, difficulties and frustrations. This thesis is my special gift to him and I believe he would be impressed. My love goes to both my parents.

Publications

This thesis is based on the following publications:

- Zhao Geng, Zhenmin Peng, Robert S.Laramee, Rick Walker, and Jonathan C. Roberts, **Angular Histograms: Frequency-Based Visualizations for Large, High-Dimensional Data**, *IEEE Transactions on Visualization and Computer Graphics (IEEE TVCG)*, Vol. 6, No. 12, December 2011, pages 2572-2580

- Zhao Geng, Robert S. Laramee, Fernando Loizides, and George Buchanan, **Visual Analysis of Document Triage Data**, *International Conference on Information Visualization Theory and Application (IVAPP)*, pages 151 - 163, Vilamoura, Algarve, Portugal, March 5-7, 2011

- Zhao Geng, Robert S. Laramee, David M. Berry, Alison Ehrmann, and Tom Cheesman, **Visualizing Translation Variation: Shakespeare's Othello**, *Advances in Visual Computing, Lecture Notes in Computer Science LNCS*, Volume 6938 (Proceedings of the 7th International Symposium on Visual Computing (ISVC), 26-28 September, 2011, Las Vegas, NV) pages 653-663, Springer

- Zhao Geng, James Walker, and Robert S. Laramee, **Markov Chain Driven Multi-Dimensional Visual Pattern Analysis with Parallel Coordinates**, *In Proceedings of Vision, Modeling, and Visualization (VMV)*, 12-14 November 2012, Magdeburg, Germany, pages 191 - 198

- Zhao Geng, Robert S. Laramee, Kevin Flanagan, Stephan Thiel, and Tom Cheesman, **Visual Analysis of Segment Variation of German Translations of Shakespeare's Othello**, Technical Report, Department of Computer Science, University of Wales, Swansea, UK, 2012, *Accepted in Information Visualization Journal with Minor Revision*

- Zhao Geng, Robert S.Laramee, Tom Cheesman, Andrew Rothwell, David M. Berry, and Alison Ehrmann, **Visualizing Translation Variation of *Othello*: A Survey of Text Visualization and Analysis Tools**, Technical Report, Department of Computer Science, University of Wales, Swansea, UK, *Under Review*

- Zhao Geng, Richard Walker, Serban Pop, Robert S. Laramee, and Jonathan C. Roberts, **Force-Directed Parallel Coordinates**, Technical Report, Department of Computer Science, University of Wales, Swansea, UK, March 2011, *Under Review*

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

Contents

1.1 Preliminaries: Data Visualization

Thanks to the advances of automated data collection tools in a large number of application domains, such as remote sensing, bioinformatics, simulation, modelling and etc., we have witnessed the explosive growth of data. As the information generated from these sources becomes larger and larger, we are drowning in the abundance of the data, but starving for knowledge. How can we effectively and efficiently extract interesting non-trivial, implicit, previously unknown and potentially useful patterns or knowledge from huge amount of data in order to help the researchers from other domain better understand their experimental result, plays a crucial role in today's scientific research. Other than some conventional approaches, such as statistics, data mining and pattern recognition, in this thesis we focus on how to utilize visualization approach to address these challenges for large data analysis.

As defined by Matthew Ward and et al. [WGK10], visualization is the communication of information using graphical representations. This definition, as we interpret communicates two important properties of data visualization. The first property is graphical representation, which maps visual metaphors to data elements in order to present, convey and analyze the important characteristics and features of the data samples, such as trends, clusters, outliers and correlations. The power of this graphical representation compared to traditional data analysis such as statistics, lies in its capability to help the observer form a mental model or mental image of something which can be rapidly and effectively recognized and understood [Spe07]. The second important property of data visualization as implied by Matthew Ward and et al. [WGK10] is the communication of information. With respect to communication, only relying on a single view of a graphical representation may not be enough, unless humans themselves are actually involved in this communication such that they are able to directly interact with the presented

visual information. As defined by Ji Soo Yi and et al. [YaKSJ07], interaction within the data and visualization context is a mechanism for modifying what the users see and how they see it. Combining the graphical representation and interaction support, we can think of visualization as a human cognitive activity. Data, in whatever form, is transformed into pictures, and the pictures are interpreted by a human being [Spe07]. Knowledge in the end is discovered or retrieved by the interaction between the human being and images.

1.1.1 Parallel Coordinates

The main technique we utilize is called parallel coordinates. Parallel coordinates, introduced by Inselberg and Dimsdale [Ins09, ID90b], is a widely used visualization technique for exploring large, multi-dimensional data sets. To show a set of points in an n-dimensional space, a backdrop is drawn consisting of n parallel lines, typically vertical and equally spaced. A point in n-dimensional space is represented as a polyline with intersection on the parallel axes; the position of the vertex on the i^{th} axis corresponds to the i^{th} coordinate of the point. Parallel coordinates is powerful in revealing a wide range of data characteristics such as different data distributions and functional dependencies [KK96]. However, one of their inherent limitations has to do with the rendering of data sets with large volume and high dimensionality.

1.2 Challenges

Ben Shneiderman [Shn96] proposed the visual information seeking mantra: overview first, zoom and filter and details on demand, as visual design guidelines for interactive information visualization application. However, this knowledge discovery process is often hampered when rendering large and high-dimensional data sets, because these data sets often cause a cluttered visualization which makes it difficult for a user to understand an overview of the data. If the user is unable to get a clear overview, it may become infeasible for them to determine which parts of the data can be filtered or zoomed in for more detail. In this section, we introduce two major challenges for multidimensional data visualizations we address in this thesis.

Figure 1.1: This figure shows the original parallel coordinates on animal tracking data.

1.2.1 Data Sets with Large Volume

In addition to the cluttered overview, a large data set slows interaction, making the data exploration process laborious.. Therefore it is important to efficiently generate an information-rich overview of large data sets and enable a fast interaction process for the user. In chapters 2 and 3, we have proposed two algorithms to overcome the over-plotting problem. Our techniques are applied to the analysis of animal sensor data. Biologists at Swansea university have collected large amounts of data relating to animal movement by attaching sensors to individual subjects. The data here was captured at 8Hz for 8 hours and 40 minutes. In this work, we selected 10 important data attributes which result in 1,048,566 records. The data attributes include: two accelerometers attached on the animal recording the acceleration parameters in X, Y and Z directions and an environment sensor recording the temperature, light-intensity (Infra depth) and pressure from the outside environment. The original data set can be plotted on parallel coordinates, but suffers from overplotting as shown in Figure 1.1.

1.2.2 Data Sets with High Dimensionality

The type of text visualization as we focus in our work, is not about typography, layout, and figurative diagrams. It is about representing the underlying semantic and structure of a text or a group of texts. In order to extract the features from the original text, as defined by James Wise and et al. [WTP+95], there are three main orders of statistics: First-order statistics. Based on word counts, such as word frequencies, concordances ; Second-order statistic: Clustering algorithms to identify related documents or word patterns ; Third-order statistic: Identify semantics through natural language understanding algorithms. In this thesis, we focus on the first and second-order statistics. Because such statistics often suffer from high-dimensionality, this brings challenges to present it in a limited screen space. Chapters 4, 5 and 6 describe our approach to handle these high-dimensional textual metadata. Our visualization system is applied to the study of German Translations of *Othello*. The domain experts from Arts and Humanities have collected 57 different German translations of Shakespeare's play, *Othello*. For each translation, metadata recorded includes the author name, publication date, country, title of the play and impact index. The translations were written between 1766 and 2006 in seven different countries including Germany (pre-1949), East Germany (1949-1989), West Germany (1949-1989), FRG (Germany since 1989), Austria, Switzerland and England. The impact index refers to each translator's productivity and reputation. it includes the re-publication figures or each *Othello* translation. Figures were derived from the standard bibliography of Shakespeare in German [HS03]. The index has five levels ranging from 1 to 5, where 1 means that the translator is not listed in the bibliography and 5 means that more than 50 publications and re-publications by the translator are listed in the bibliography.

1.3 Proposed Solutions : Overview and Contribution

In this thesis, we aim to address two intrinsic challenges in multidimensional data visualization, namely presenting and analyzing large and high-dimensional data sets. For each of challenge, we have applied our proposed techniques to a real word data set and work closely with domain

experts. This section contains summaries for the following main chapters and states our main contributions in this thesis. Chapters 2 and 3 described solutions to the first challenge, namely visualization of large volume data. Whereas chapters 4,5 and 6 introduced our solutions to the second challenge, namely visualization of high-dimensional data. For continuity, these summaries are arranged to start with corresponding chapter headings.

1.3.1 Visual Analysis of Document Triage Data

As part of the information seeking process, a large amount of effort is invested in order to study and understand how information seekers search through documents such that they can assess their relevance. This search and assessment of document relevance, known as document triage, is an important information seeking process, but is not yet well understood. Human-computer interaction (HCI) and digital library scientists have undertaken a series of user studies involving information seeking, collecting a large amount of data describing information seekers' behaviour during document search. Next to this, I have witnessed a rapid increase in the number of off-the-shelf visualization tools which can benefit document triage study. Here I set out to utilize existing information visualization techniques and tools in order to gain a better understanding of the large amount of user-study data collected by HCI and digital library researchers [GLLB11].

1.3.2 Angular Histograms: Frequency-Based Visualizations for Large, High Dimensional Data

In this chapter, I propose a novel solution, namely, angular histogram to address the over-plotting caused by rendering large data sets using parallel coordinates. This technique is a frequency-based approach to large, high-dimensional data visualization. It is able to convey both the density of underlying polylines and their slopes. It offers an intuitive way for the user to explore the clustering, linear correlations and outliers in large data sets without the over-plotting and clutter problems associated with traditional parallel coordinates. I demonstrate the results on a wide variety of data sets including real-world, high-dimensional biological data. Finally, I compare my method with the other popular frequency-based algorithms [GLC+11].

1.3.3 Markov Chain Driven Multi-Dimensional Visual Pattern Analysis with Parallel Coordinates

In chapter 3, I have proposed frequency-based approach using binning and histograms for clutter reduction in parallel coordinates. The traditional binning method, which records line-segment frequency, only considers data in a two-dimensional subspace, as a result, the multi-dimensional features are not taken into account for trend and outlier analysis. Obtaining a coherent binned representation in higher dimensions is challenging because multidimensional binning can suffer from the curse of dimensionality. In this chapter, I utilize the Markov Chain model to compute an n-dimensional joint probability for each data tuple based on a two-dimensional binning method. This probability value can be utilized to guide the user for selection and brushing. My system provide various interaction techniques for the user to con-

trol the parameters during the brushing process. Filtered data with a high probability measure often explicitly illustrates major data trends. In order to scale to large data sets, I also propose a more precise angular representation for angular histograms to depict the density of the brushed data trends. I demonstrate my methods and evaluate the results on a wide variety of data sets, including real-world, high-dimensional biological data [GWL12].

1.3.4 Visualizing Translation Variation of *Othello* : A Survey of Text Visualization and Analysis Tools

Recognized as great works of world literature, Shakespeare's poems and plays have been translated into dozens of languages for over 300 years. Also, there are many re-translations into the same language, for example, there are more than 60 translations of *Othello* into German. Every translation is a different interpretation of the play. These large quantities of translations reflect changing culture and express individual thought by the authors. They demonstrate wide connections between different world regions today, and reveal a retrospective view of their cultural, inter-cultural, and linguistic histories. Researchers from Arts and Humanities at Swansea University are collecting a large number of translations of William Shakespeare's *Othello*. In recent years, since circa 2005, I have witnessed a rapid increase in the number of off-the-shelf text visualization tools which can benefit the study of translations of *Othello*. Here I set out to survey and utilize existing text visualization techniques and tools in order to gain a better understanding of the various translations of Shakespeare's work. In this chapter, I describe and compare various freely available text visualization software tools. In the next two chapters, I will introduce our solutions for visualizing the variations of German translations of *Othello* [GSC+12].

1.3.5 Visualizing Translation Variation on Term Level: Shakespeare's *Othello*

In this chapter, I have developed an interactive visualization system to present, analyze and explore the variations among these different translations on a document term level. My system is composed of two parts: the structure-aware Treemap for document selection and meta data analysis, and Focus + Context parallel coordinates for in-depth document comparison and exploration. In particular, the domain experts want to learn more about which content varies highly with each translation, and which content remains stable. They also want to form hypotheses as to the implications behind these variations. My visualization is evaluated by the domain experts from Arts and Humanities [GLC+11].

1.3.6 Visualizing Translation Variation on Segment Level: Shakespeares *Othello*

In chapter 5, I have introduced my system to visualize the translation variation at the level of user-defined terms. In this chapter, I have described an interactive focus+context visualization system to present, analyze and explore variation at the level of user-defined segments. From this visualization, the domain experts are able to obtain an overview of the relationships of similarity between parallel segments in different versions. I can uncover clusters and

outliers at various scales, and a linked focus view allows us to further explore the textual details behind these findings. The domain experts who are studying this topic have evaluated my visualizations and I have reported their feedbacks. My system helps them better understand the relationships between different German translations of *Othello* and learn some new insights [GLF+12].

1.3.7 Appendix : Force-Directed Parallel Coordinates

In this appendix, I present force-directed parallel coordinates, a technique that uses a familiar visual metaphor to represent data in PCPs by modelling the plot as a physical system, together with some physical interactions with the user.

Chapter 2

Visual Analysis of Document Triage Data

Contents

2.1 Introduction

This chapter is based on a publication from Geng et al [GLLB11]. Document triage is an important stage of the information seeking process. It focuses on user behavior with respect to skimming, evaluating and organizing documents when searching for information. Various studies have been conducted [BBM+06, BL07, LB09] to explore users' behaviour during document triage. Over the course of these studies, a large amount of qualitative and quantitative data is collected. However, understanding and analyzing this data is difficult in its raw form. Conventionally, these experimental data are analyzed by statistical methods and simple visualizations, such as bar charts, line graphs and pie charts. These simple visualizations are useful, but of limited help for semantically rich data. Thus there is a great demand for summarizing and presenting the data in a more insightful way that HCI scientists can better utilize. This motivates the exploitation of more advanced information visualization techniques. In recent years, we have witnessed a rapid increase in the number of visualization tools for general use, such as XMDV [War94a], Mondrian [The02, MS08], TopCat [M. 05] and ManyEyes [VWvH+07]. We refer to Gilson [Gil08] for an overview of tools. We carry out an investigation on how

well data collected by HCI and digital library researchers can be visualized by existing off-the-shelf information visualization tools and how well each can be applied. Results show that the amount of time spent on documents, pages and document features as depicted by some of our visualizations, such as the treemap, parallel coordinates, stack graph, matrix chart and 3D bar chart can help HCI and digital library scientists understand and explore user behaviours during document triage. On the other hand, we also learn that some of the visualizations, such as the 2D bar chart, 2D and 3D scatterplot are limited in their applicability to this problem. The advantages and disadvantages of the most promising visualization techniques are compared and evaluated.

The aim of this chapter is not to present new visualizations techniques or interactive tools. There are three main goals. The first goal is that the HCI experts share their user study data with visualization specialists and specify their exploratory requirements. The second goal is for the visualization group to then use the raw data produced by the HCI scientists to create a selection of visualizations which were not currently used/known by the HCI scientists. In turn the HCI scientists would be able to analyze the produced visualizations and comment on their usefulness and applicability in their research, The third goal is the specification for the potential for creating specialized tools for HCI and Digital Librarian researchers.

In this chapter we contribute the following:

- A novel attempt to systematically visualize experimental document triage data studying human behaviors using state-of-the-art information visualization methods.
- We survey the variety of state-of-the-art, off-the-shelf information visualization tools in order to help researchers from another domain gain insight into their experimental data.
- We compare and evaluate the various tools and visualizations with respect to their effectiveness in solving a given problem.
- The results of our investigation are evaluated by HCI and digital library researchers studying document triage.

The result of our study also provides the reader with a concise introduction to free, off-the-shelf information visualization applications and their features.

The rest of this chapter is organized as follows. In section 2.3 I briefly review the past and related work in the study of document triage. In section 2.4, I describe in detail the data collected during the document triage study. In Section 2.5 and the supplementary material, I develop an objective metric to rate document relevance and compare it with the user's subjective relevance score. In section 2.6, I investigate different visualization techniques from various tools and evaluate the usefulness of each. Section 2.7 presents a subjective comparison and evaluation of the usability of visualization softwares. Section 2.8 contains feedback from the domain experts studying this problem.

2.2 Exploratory Specifications

The data provided by the HCI group, as discussed in Section 2.4 is quantitative in nature, including timings and numerical ratings from participants. The main aim of a visualization for

the HCI researchers is to give a fast overview of the data in order to formulate hypotheses on a) relationships between document properties, times and ratings and b) common recurring patterns over all the three areas mentioned in part (a). These can then be tested empirically for validity by statistical significance. Thus far, hypotheses are inferred before the study by previous results or by observing the individual behavior of participants as they perform a specific task. Indeed, many hypotheses are speculative and are sometimes based on curiosity rather than evidence. We hope that visualizations will greatly decrease the time taken to formulate more grounded hypotheses and dismiss non substantive data patterns. Furthermore, we hope to be able to test for patterns and relationships which may have previously gone unnoticed without visualization of the information.

2.3 Related work

Current research on document triage is heavily focused on user-studies and laboratory-based observational experiments with respect to user behaviours. C. Cool et al. conducted two studies on students and scholars, to investigate factors which underly readers' judgements of the document relevance to their particular information need [CBFK93]. R. Badi et al. designed a user-study to recognize user interest and document value from reading and organizing activities in document triage [BBM+06]. Buchanan and Loizides [BL07] conducted an experiment designed to compare users' initial relevance decisions with respect to paper and electronic media. Afterwards they launched a controlled empirical study on users' activities during document triage [LB09]. They hypothesized that some document features, such as abstracts, titles, and conclusions strongly influence user's search patterns. Visual document features, such as pictures, had a mild effect on subjective relevance ratings.

Up to now, various visual analysis tools for document triage have been developed. The NIRVE [SVM+99] provides a 3D interface for an overview of document sets and details of individual documents. Trist is a customized research prototype that helps users navigate through thousands of documents resulting from a search query [JWS+05]. The Tag cloud tools [BGN08] highlight the important words in web documents and aid users in relevance decisions.

The customized research prototypes described above are all useful in helping users triage through a large number of documents. The focus of our work is the study of the document triage process itself. We visualize the data from experiments that observe the behavior of information seekers as they search through documents. This is the process that takes place after an initial search engine result has been given. The goal is to discover which document features influence readers the most during document triage.

There are many general purpose information visualization tools developed for industry, such as Eureka, SpotFire and InfoZoom [Kob01]. They provide various interactions for users to follow the "Visual Information Seeking Mantra" : overview first, zoom and filter, details on demand [Kei02, Shn96]. Advanced tools such as OpenViz [ADV], ILog Discovery [BT04] and Tableau [HMSA08], are integrated with multiple visualization techniques to handle complex data sets and queries. These tools are for commercial use. Our study focuses on free, off-the-shelf visualization software, because they are easily and freely accessible for researchers from

other domains. As stated by Kobsa [Kob01], when solving a specific problem, users, especially from other domains, might have great difficulties in selecting the most effective visualizations out of numerous choices. Also, the task questions during the user study, effect how the user derives information from a visualization [ZK08]. It is important to note that although there have been several general user-study evaluations of information visualizations [Kob01, ZK08]. The work presented here is not a general user-study but a very specialized investigation for a focused audience, namely, document triage researchers. Although we do believe the work conducted here can benefit other users as well. Our work is to facilitate document triage experts search for visualizations that give more benefit for their experimental data.

2.4 Background : User-Study Data

The user-study data applied in our visualization, was collected by our HCI collaborators, Buchanan and Loizedes. Their experiment [LB09] aims to investigate human behaviours in the process of reading documents, searching for information and evaluating document relevance. During the study, 20 participants performed document triage on a closed corpus of electronic PDF documents, evaluating each for its suitability for two tasks. LOG data of their interactions is captured. The documents provided for search range from short papers (2 pages) to full journal papers (29 pages). There are 6 documents including TABLET to TABLET5 in Task 1, and 10 documents including HCI to HCI9 in Task 2. In Task 1, the goal is to find material on the interfaces of tablet PC's. In Task 2, the goal is to find papers on specific CHI evaluation methods. The participants' ages range from 21 to 28. They are studying at postgraduate level in the computer science discipline, and all have experience with PDF reader software. In this section we briefly describe the data collected during the study.

1. Pre-study questionnaire: Study participants filled out a questionnaire before the experiments indicating their: (1) age, (2) number of years of experience of using electronic document readers, (3) average number of academic documents triaged per day, (4) average amount of time per day spent searching documents. Participants were also asked to rate the importance of the following document attributes in a range from 1 to 10 (1 meaning "very irrelevant" and 10 meaning "very relevant"): main title, headings, introduction, plain text, conclusion, references, images and figures, highlighted and emphasized text. Participants also indicated their preference for searching on paper versus using a computer.

2. Data recorded for each participant during the study: For each participant the total amount of time in 1/5th of a second accuracy viewing each page of each electronic document was recorded. From this the total time viewing each document can be calculated. There are 10 types of document features appearing in the study. We abbreviate the document features in order to optimize the space for visualizations, as shown in Table 2.1. During the study, participants' viewing time on pages is logged. The eye-trackers are not available during this user-study, thus the viewing time on document features can only be inferred based on the HCI researchers' hypothesis that more time spent on a page suggests more interest on the features on that page, as shown in equations (2.1) and (2.2). The frequency of document feature appeared in each page can be defined as:

Number	Features	Abbreviation
1	Heading	He
2	Abstract	Ab
3	Keywords	Kw
4	General Term	Gt
5	Emphasized Text	Em
6	Figure	Fi
7	Conclusion	Co
8	Reference	Re
9	Picture	Pi
10	Plain Text	Pl

Table 2.1: Abbreviation of document features. Plain text means a page contains none of features from 1 to 9. Emphasized text includes bullet point, bold text, italic text and underlined text. The abstract, keywords and general terms are features here, but not included in previous literature [LB09].

$$P_{i,k,q} = \frac{n_{i,k,q}}{N_{k,q}} \tag{2.1}$$

where $n_{i,k,q}$ represents the number of document feature i appeared in page k of document q, and $N_{k,q}$ represents the total number of all features appear in page k of document q.

Thus the viewing time for each feature can be estimated as:

$$t_{i,k,q} = T_{k,q} \times P_{i,k,q} \tag{2.2}$$

where $T_{k,q}$ represents the participants' average viewing time on page k of document q.

We conclude that the viewing time for a document feature is determined by two factors : the frequency of this document feature in a page and the participants' viewing time on that page. The HCI researchers wanted to see how such factors would potentially influence participants' reading patterns by the use of the visualizations. The new findings or hypotheses obtained then will be used for further experimental design aided by eye-trackers.

3. Participant rating of document relevance: After each search task participants were asked to assign each document a relevance score (1 meaning"worst" or very irrelevant and 10 meaning"best" or most relevant). Also, for each document, the significance of the following document features was recorded: (1) headings (2) picture (3) figures (4) emphasized text.

2.5 Objective Relevance Metrics

As part of this investigation we attempt to derive some objective document relevance metrics for the documents involved in the triage study. These objective metrics may then be used to gain insight into how effective participants are in their search for relevant information and can also be compared with subjective metrics. We use the term *corpus* to refer to the whole

document set, *query* to refer to the target information users are requested to look for and *term* for a unique word in the document.

Most existing information retrieval (IR) systems utilize a numerical score to grade the document relevance and rank documents by this score. The most popular model for this process is called the vector space model [KJ02, WLWK08, LCS97]. In this model, the list of terms associated with their weight is treated as the document vectors. The weight of each term indicates the importance in a document, and is determined by $Tf \times Idf$.

Tf (Term Frequency) is simply the number of times a term occurs in a given document, but it's often normalized by dividing the total number of times all terms appear in the given document. It's a measurement for the importance of a word in a given document, and can be defined as:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $n_{i,j}$ is the number of times a word t_i appears in document d_j. The denominator is the total number of occurrences of all terms in the document.

Idf (Inverse Document Frequency), as its name implies, is the inverse of the Document Frequency. The Document Frequency is the number of documents a word occurs in within the corpus, here corpus refers to the whole collection of documents. The IDF model often takes the logarithm of the Inverse Document Frequency, to measure the general importance of a term. It can be defined as:

$$idf_i = log\frac{|N|}{|d : t_i \in d|}$$

where $|N|$ is the total number of documents in the corpus, $|d : t_i \in d|$ is the number of documents the word t_i appears in.

Thus the weight of a term i in document j can be defined as

$$w_{i,j} = tf_{i,j} \times idf_j = tf_{i,j} \times log\frac{N}{df_i}$$

where N is the total number of documents in the corpus, *df* is the document frequency and *idf* is the inverse document frequency. Large values of $w_{i,j}$ imply term i is an important word in document j but not common in all documents N.

The similarity of document D_i to a query Q is the total weight of all key terms. It determines the objective relevance of the document.

Spink and Jansen et al [JSS00, SJWS02] observe that when users search information on the web, their queries are short, such that about two in three have one or two terms, and less than 4% of the queries contain more than 6 terms. Considering this, each of our task queries can be shortened and highly abstracted into shorter key terms, such as"Teaching Tablet PC" and "Touch Screen" from Task 1, and "Evaluation Techniques" and "Product Design" from Task 2. These key terms are chosen rather subjectively for our purposes. The porter stemming is used to include their plural, -ing, and -ed forms, thus keywords "Teaching" and "Teacher" will all be reduced to "Teach", as shown in Tables 2.2 and 2.3. From these two Tables, the number of occurrences for key words in every document is recorded and the total number of occurrences

of terms in each document is counted. Furthermore, we compute the total *Tf* and *Idf* value of the key words for each document.

	Teach	Tablet	PC	Touch	Screen	Total Words	Total TF	Total IDF
TABLET	3	28	21	0	0	3812	0.014	4.329
TABLET1	3	35	36	0	11	5334	0.016	6.829
TABLET2	16	71	67	0	11	5251	0.031	6.829
TABLET3	5	21	16	0	0	3323	0.013	2.942
TABLET4	2	65	56	0	9	3319	0.04	3.769
TABLET5	8	27	12	1	0	11261	0.004	6.003

Table 2.2: TF-IDF value of selected key words from Task 1. The frequency of key words is recorded. Document TABLET4 receives the highest *TF* score although it contains fewer key words than TABLET2. Document TABLET5 is the longest with lowest *TF* score.

	Evaluation(s)	Design(s)	Product(s)	Technique(s)	Total Words	Total TF	Total IDF
HCI	1	19	0	4	9276	0.003	2.735
HCI1	0	14	0	0	439	0.032	0.352
HCI2	56	54	12	2	7741	0.016	4.003
HCI3	10	23	0	3	7467	0.005	1.284
HCI4	1	59	1	3	5954	0.011	3.31
HCI5	1	4	3	2	603	0.017	2.082
HCI6	0	19	0	0	1168	0.016	0.822
HCI7	17	30	7	9	7537	0.008	7.287
HCI8	3	12	4	0	612	0.031	4.545
HCI9	0	18	3	0	569	0.037	1.621

Table 2.3: TF-IDF value of selected key words in Task 2. The frequency of key words is recorded. The document HCI is the longest document with lowest relevance by total TF and the document HCI9 receives highest TF score but is not the most relevant document in Task 2 due to its low IDF score, as shown in Table 2.4.

In contrast to the objective relevance score, we collect the subjective relevance rating for each document from the participants during the user study. Participants assess the relevance of the document in a range from 1 to 10 (1 meaning"least relevant", 10 meaning"most relevant"). We normalize the scores in the range 0 to 1 to compare with the objective metrics, as shown in Table 2.4.

The comparison of objective and subjective document relevance metrics can be visualized using the bubble chart and line graph respectively, as shown in Figure 2.1 and 2.2. By simply viewing the bubble chart visualization the domain expert are mistakenly led to think that due to the unevenness of the size of the bubbles, that there is no correlation between the occurrence of popular terms in the documents and the participant rankings. Note that this is a test that was not performed when looking at the data without the help of visualizations. They were surprised to then observe the line graph visualization in Figure 2.1 which revealed a relationship between term occurrence and document ratings. It seems that, although the sizes of the subjective bubbles in the first visualization were a different size than the objective bubbles, the size proportion between the corresponding documents in each category has a positive correlation. We can see from this that the bubble visualization would be useful to our work when

Corpus	Documents	Objective	Subjective
Task1	TABLET	0.194	0.67
	TABLET1	0.184	0.605
	TABLET2	0.39	0.74
	TABLET3	0.179	0.605
	TABLET4	0.639	0.585
	TABLET5	0.034	0.53
Task2	HCI	0.006	0.63
	HCI1	0.031	0.51
	HCI2	0.053	0.695
	HCI3	0.009	0.555
	HCI4	0.016	0.57
	HCI5	0.208	0.61
	HCI6	0.062	0.325
	HCI7	0.044	0.69
	HCI8	0.544	0.66
	HCI9	0.178	0.605

Table 2.4: This table shows the objective and subjective ratings for each document in Task 1 and Task 2. Both scores are normalized between 0 and 1. TABLET4 is the most relevant document using objective metrics, whereas TABLET2 is the most relevant according to subjective score. The documents are ranked by the document order in each corpus.

comparing two groups, but care needs to be taken to match the correct type of variables for the two groups. The best application would be for the researcher to get an overview of ratings between two or more groups but with the same criteria for a task. Due to this limitation and risk of misinterpretation of the data it is deemed quite difficult to apply the bubble chart visualization to effective exploratory research in our work. We are however, convinced that the line graph visualization can produce much more accurate overviews of relations between groups and patterns.

2.6 Visualization

In this section, we utilize various existing visualization techniques and tools to investigate document triage data. The following list summarizes the tools and their visualizations we have tried:

- The ManyEyes [VWvH+07] application with the following visualizations: Wordle, Tag Cloud, TreeMaps, Line Graph, Stack Graph, Bar Chart, Bubble Chart, Scatterplot, and Matrix Chart
- The XMDV [War94a] application with the following: Parallel Coordinates, Scatterplot Matrix, Star Glyphs and Dimensional Stacking
- The Mondrian [MS08, The02] application with the following visualizations: Bar Charts, Histograms, Parallel Coordinates, Boxplots, Scatterplot Matrix
- The Treemap Application 4.1 [Kob04]: TreeMaps
- The Topcat Application [M. 05]: 3D Scatterplots, Histogram, Sky, Lines and Density

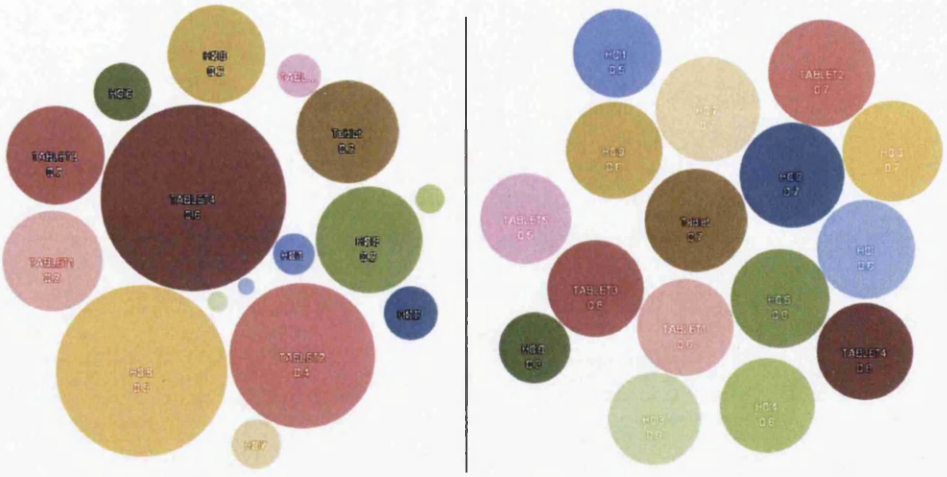


Figure 2.1: We can plot the subjective and objective ratings described in Section 2.4 and 2.5 onto the bubble chart in ManyEyes [VWvH+07]. As shown in the figure, the top one presents the objective ratings. Whereas the bottom one showing the subjective ratings, we can observe that the difference of scores between documents is very slight. This visualization highlights the discrepancies between the objective and subjective relevance metrics.

- Microsoft Office 2007 [Mic07]: 3D Barchart, Radar chart, 3D line graph, 3D Bubble Chart. Although Excel is a commercial application, we include it as an exception because we have a university license for this product.
- The Tableau Application [HMSA08]: The free trial version only contains a few basic visualizations. Advanced options such as parallel coordinates are not available for use in the free trial

ILog Discovery [BT04] and OpenViz [ADV] are advanced information visualization applications, but they were not available for downloading during the course of our study.

In the following subsections, we describe the applications we used along with the visualizations of document triage data and evaluate the usefulness of each. We tried over 17 visualizations with several different variations for a total of 110 images. Each image is stored in its original resolution on the supplementary website `http//cs.swan.ac.uk/~cszg/docTriage`. Each tool was systematically applied to the same data described in Section 2.4 [LB09] by visualization researchers. For each tool, we (1) re-formated the data to match the application's input requirements, (2) tried out each of the visualizations offered by the tools, and (3) evaluated the utility based on the domain expert's feedback. The visualizations are assessed by domain experts - the HCI scientists who carried out the user-study [LB09]. Due to space limitations, we cannot describe every visualization we tried out, but only those most relevant and beneficial to the investigation. The beneficial visualizations are able to provide more insight for the data set and help the HCI researchers obtain and form new findings and

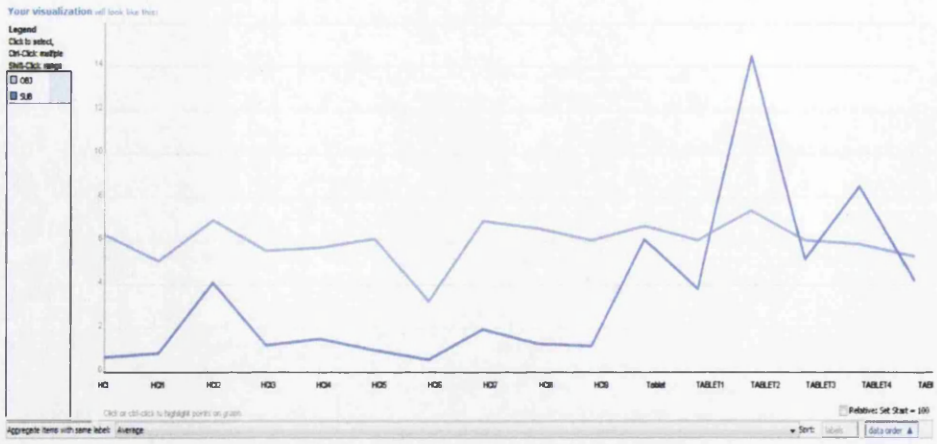


Figure 2.2: This figure shows the line graph plot of the subjective and objective rating scores. Documents in the order from HCI to HCI9, TABLET to TABLET6 are mapped to the x-axis. We can observe that, except documents HCI1, HCI7, HCI8 and TABLET4, the subjective rating (the line above) and the objective score (the line below) correspond in a linear fashion.

hypothesis, as specified in Section 2.2. We also provide some of the less beneficial visualizations as supplementary material. Some of the less beneficial visualizations include: bar charts, bubble charts, 2D and 3D scatterplots.

2.6.1 Stack Graph Visualization

The stack graph in ManyEyes is used to visualize the total change of a group of quantities over time [VWvH+07]. During the document triage study, HCI scientists observe that a user's triage process can proceed in a linear fashion starting with the first document and then reading and scoring every subsequent document [BL07]. The sequence of documents in Task 1 and 2 can be mapped to the time parameter of the stack graph. For each document, we can observe the changes in viewing time spent on individual pages from top image in Figure 2.3. This visualization shows participants spent most of their time viewing page one. Also, users spent less time on pages near the end of the documents. From the peak of each document, we can rank the documents by viewing time, e.g. HCI receives the most time and HCI(6) the least. Furthermore, we can compare individual pages of different documents, such as all pages of TABLET(2) receive more viewing time than adjacent documents.

The 2D stack graph utilizes the accurate graphical perception encodings [CM85], such as position, length, area, angle slope and color, to convey multiple data attributes to the user simultaneously. Also, an additional variate, namely, documents, can be included in the visualization as opposed to just two dimensions (page and time) in the bar chart, line graph or pie chart. However too many pages in this visualization leads to problems such as very thin strips or degenerate line strips. It's difficult to discern the last few pages of longer documents (10 pages more), thus the length of such documents is difficult to infer.

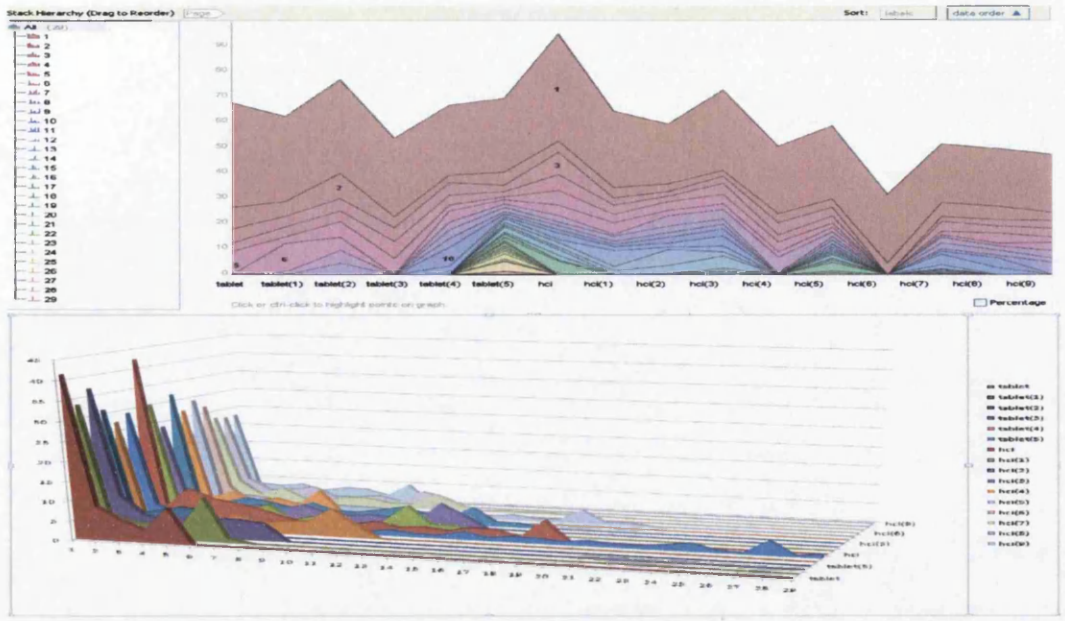


Figure 2.3: The top image shows the 2D stack graph visualization of document triage data in ManyEyes [VWvH+07]. The X-axis represents documents in both Task 1 and Task 2. Y-axis represents the average viewing time on each page over all participants in each document. The strips in different colours represent viewing time trends for individual pages. The number in every strip indicates the page number. The bottom image illustrates the 3D stack graph plotted with Microsoft Office Excel 2007. The X-axis is mapped to the page number, Y-axis to the viewing time and Z-axis to the documents. Compared with the 2D stack graph, we can gain an overview of all documents' time and page distribution and compare them more intuitively.

The problems of degenerate or overlapping strips can be reduced in the 3D stack graph, which is available in Microsoft Office Excel 2007, as shown in Figure 2.3 bottom. Compared with the 2D stack graph, the length of each document is clearly shown in 3D space. Also, we can gain an general trend of participants viewing time on documents and pages which 2D stack graph cannot offer. Although the 3D stack graph suffers from occlusion and perspective distortion [Shn03], the domain expert feels that the benefits provided by 3D outweigh the drawbacks in this particular case.

2.6.2 Treemap

HCI researchers study how document features influence user behaviors when searching documents. The relationship of viewing time between pages and document features may unveil user navigation patterns during document triage. In order to optimize the space to display more information, we abbreviate the nodes in the tree structure. Pg1, Pg2 etc. are page numbers. TA, TA1 etc. and HC, HC1 etc. represent documents in Task 1 and 2. The abbreviation of document features is given in Table 2.1. Figure 2.4 shows the hierarchical tree structure. Each

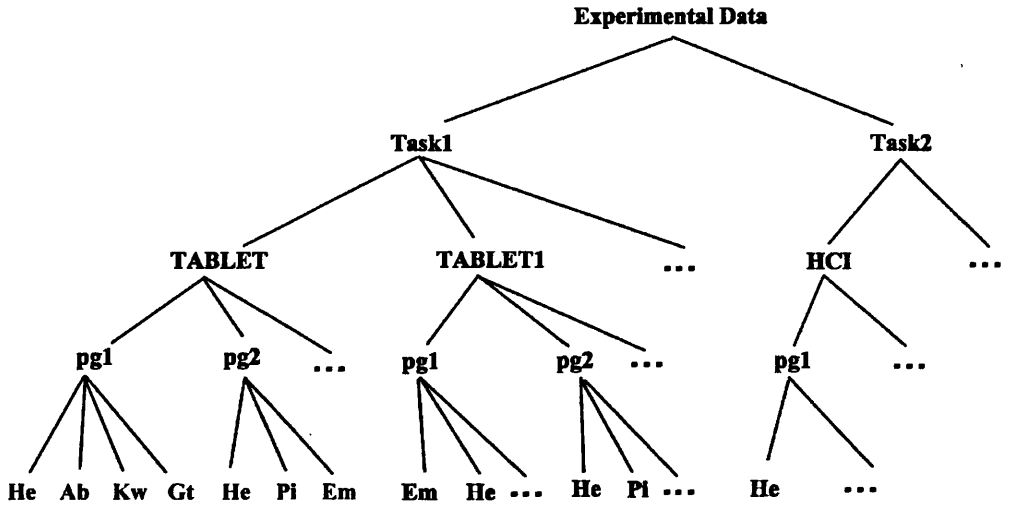


Figure 2.4: This figure illustrates the hierarchy of document triage data we used to create some of the visualizations. Document features form the leaf nodes.

page includes features, such as headings, abstract, pictures, etc. Each feature is associated with the average viewing time from the participants. A treemap is an alternative representation of a tree diagram, introduced by Johnson and Shneiderman [JS91, Shn92]. ManyEyes offers squarified treemaps, which uses rectangles with an aspect ratio close to 1 and which functions are ordered by size [BHvW00]. It also provides various navigation such as smooth zooming, hierarchy reordering and color mapping for users to interact with different levels of the tree structure.

We can create the treemap using a Task-Document-Page-Feature-Time hierarchy. We annotate the document names of treemap visualization result to indicate the intermediate nodes, as shown on the left in Figure 2.5. From this visualization, page one including its most frequent features, such as abstract (Ab), keyword (Kw) and headings (He), covers the most area in all documents except "HC6". Participants almost even out the distribution of their viewing time on each document in Task 1 and on some groups of documents in Task 2, even though the documents' length varies from 5 to 29 pages. We can hypothesize that participants' viewing time is mostly effected by the the first page, not by the document length. With the treemap, only one level in the tree structure can be displayed each time. To compare different variates, we need to frequently switch between various tree depths, which is tedious and error-prone. In order to further explore participants' viewing patterns, we need a visualization which can combine documents, pages and features together in just one view. This motivates the use of matrix chart visualization in Section 2.6.3.

The treemap Task-Document-Page-Feature hierarchy can be switched to Task-Feature-Document-Page order. Each task contains several distinct document features. Each feature appears in different documents. We manually annotate the document features, as shown on the right in Figure 2.5. From this visualization, we can observe the distribution of document

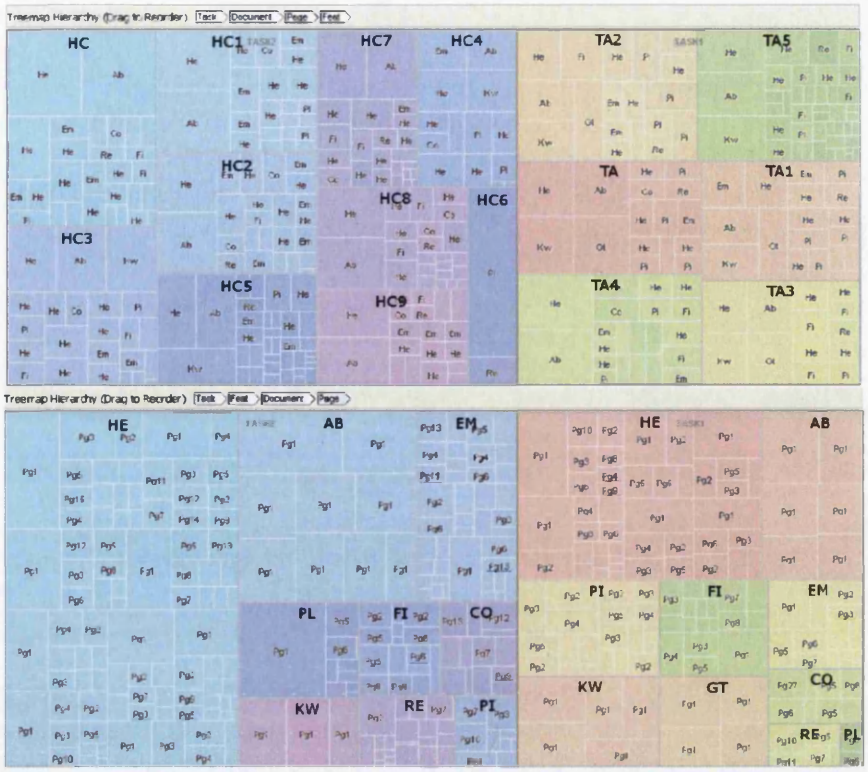


Figure 2.5: The left image shows a treemap of a Task-Document-Page-Feature hierarchy from ManyEyes [VWvH+07]. The top row of the visualization shows the current tree hierarchy. Each document name in black bold character is manually annotated. Different colors represent different documents. The document features on each page are mapped to the leaves of the tree. The right image shows a Task-Feature-Document-Page structure. Different colors represent distinct document features in both tasks. The feature names are manually annotated. The pages that features appear on are visualized as leaves of the structure.

features. Visual components, such as figures, pictures and emphasized texts, have a weaker impact than headings in terms of their population in documents and frequency in pages. Compared with Task 2, the area in plain text (Pl) in Task 1 is dramatically reduced. This is because featureless pages appear in 6 documents in Task 2, whereas only in 2 documents in Task 1. We also observe that pictures and figures in Task 1 cover much larger proportion than in Task 2. This is might because such features spread out in more pages in Task 1 than in Task 2. But this distribution of document features does not represent participants' viewing preference. In order to further explore the influence of the feature distribution, we manually calculate participants' average viewing time on documents, and pages with figures, pictures and plain texts in both tasks. The result shows that on average, participants spent more time viewing documents in Task 1 than in Task 2. Moreover, the viewing time on pages with figures and pictures is larger than on pages only contain plain texts. From these visual clues, we can form a hypothesis

that pages rich in visual features, such as pictures and figures, might draw more attention than pages only containing plain text from participants. But a treemap is not enough to borne out this hypothesis, we need more accurate user-study data focused on document features. During the analysis, we need the aggregation method of numerical attributes, for example the total and average viewing time on variates appeared in the higher level than features, such as pages, documents and tasks shown in Figure 2.4, should be calculated and displayed. Unfortunately this function is not supported for the treemap in ManyEyes.

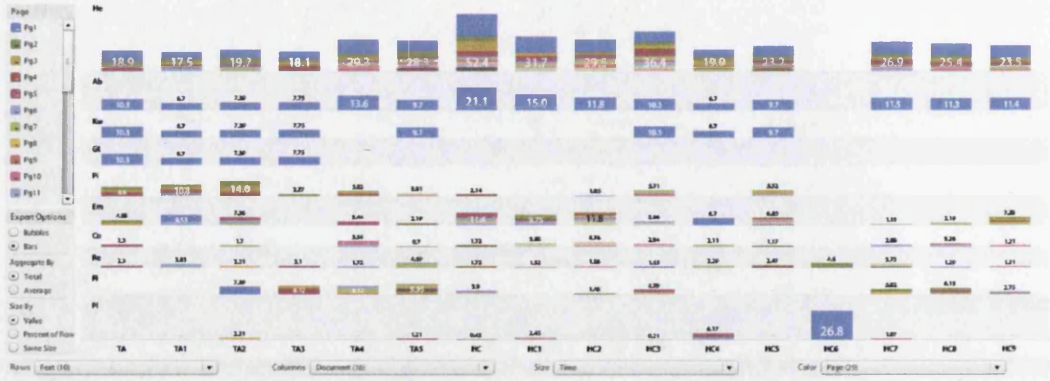


Figure 2.6: This figure shows a matrix chart in ManyEyes [VWvH+07]. Rows are mapped to the document features, columns to the documents, colors to page number and size of each bar to time. Rows, columns and colors can only accept categorical data. This visualization depicts four variates at a time and displays the general view among the four variates.

Compared with the stack graph, the treemap is able to co-relate four variates. It displays how users view document features at the per document and per page level. Also, by reordering the hierarchy of treemap with 4 variates, 24 different variations can be obtained. They present various facets of the data. But only one of the hierarchy can be displayed at a time. This presents challenges to compare two different treemap hierarchies simultaneously. This problem might be addressed by the use of coordinated multiple views (CMV).

2.6.3 Matrix Chart Visualization

Matrix chart visualization was introduced by Marsh [MS92]. It maintains tabular organization of the data, but using bars or bubbles to represent each of the elements in the table. The matrix chart is supported in ManyEyes [VWvH+07], and variates can be mapped to four visual attributes: rows, columns, size of the bubble or bar and colour, as shown in Figure 2.6. We adopt the bar chart for every row/column combination, because viewers can interpret changes in length more accurately than in area [CM85]. From this visualization, we can gain an overview of four variates simultaneously. Headings (He) is the most popular features in most documents and pages. Page one often does not contain figures and pictures. Document "HC6" jumps out as a outlier which contains the least pages and document features. We observe that document "TABLET2" is rich in visual document features, such as emphasized text (Em), pictures

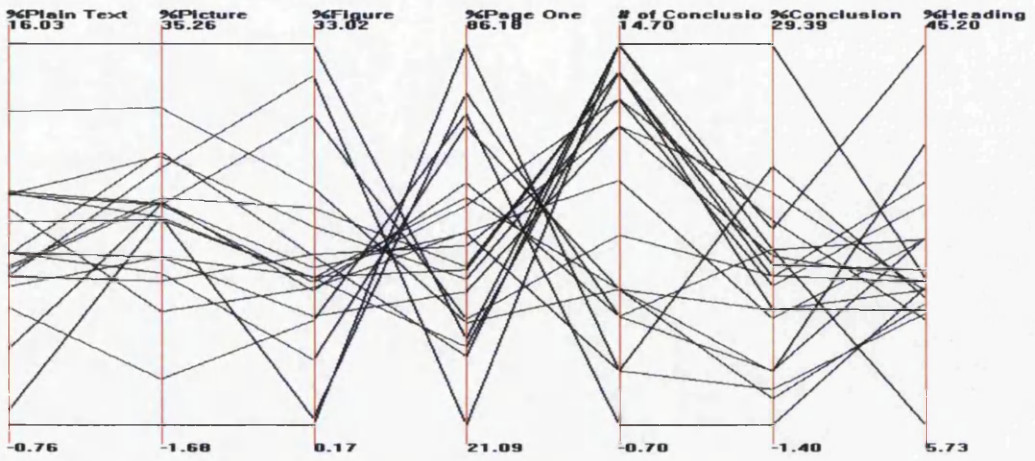


Figure 2.7: This figure shows a parallel coordinates visualization in XMDV [War94a]. From left to right, the first four axes show the percentage of each participant's viewing time on plain text, pictures, figures and page one respectively. The last three axes show the number of conclusions each participant viewed, percentage of each participant's viewing time on conclusions and headings respectively.

(Pi) and figures (Fi). Although this document only has 7 pages, it receives the second largest viewing time on average from participants. Also, we find page one in document "TABLET2" receives less viewing time than most of the other documents. This might extend our hypothesis drawn from Section 2.6.2 that participants' viewing time is not only effected by the first page, but also by the existence of visual document features in pages, such as emphasized text, pictures and figures. We also notice that pages containing a conclusion (Co) across all documents only receive little average viewing time from all participants. This seems to contradict the HCI researchers' hypothesis, which suggests that participants used to pay more attention to the document's conclusion.

A matrix chart can present the same data set as a treemap. Although it is unable to depict the hierarchies, it offers a broad view encompassing all data attributes [MS92]. Compared with the treemap in Section 2.6.2, it provides an aggregation method to calculate average and total value for numerical attributes, which offers more convenience for us to explore anomalies and patterns among four variates.

2.6.4 Parallel Coordinates Visualization

Parallel coordinates are used for displaying high-dimensional data [ID90b]. During the document triage study, for each participant, the percentage of his viewing time on pages with pictures, plain text, figures, conclusions and headings is calculated. The percentage of viewing time on page one and the number of conclusions each participant viewed are also recorded. This multivariate data can be plotted to seven axes on parallel coordinates in XMDV [War94a]. By reordering the axes, we can find several patterns of value, as shown in Figure 2.7. There are

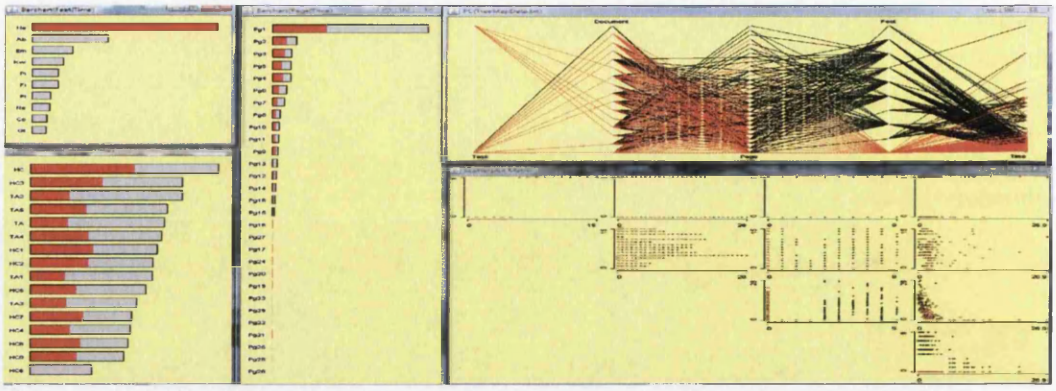


Figure 2.8: This figure shows a combination of bar charts, parallel coordinates and scatterplot matrix in Mondrian [MS08,The02]. There are five variates in the visualization: task, document, page number, document feature and viewing time. Plots are fully linked to each other. From these visualizations, we can observe the distribution of the highlighted feature heading (He) in document and page respectively.

20 polylines in the figure, each one represents every participant's reading behavior on pages with document features. From this visualization, an inverse correlation between viewing time on page one and the number of viewed conclusions is clearly revealed. This implies that as participants spent more time on page one, they are likely to overlook the conclusions, and vice versa. Also, the number of conclusions being viewed and their received viewing time reveal a correlation. All the data in viewing time on figures, pictures and page one show a general trend toward inverse correlations. It could be that page one often does not contain pictures and figures, as discussed in section 2.6.3, such that more time viewing on page one means less time is spent on figures and pictures.

Parallel coordinates are able to reveal the correlations between the viewing time of conclusions, page one, figures and pictures: observations we were unable to make with previous visualizations. A disadvantage of parallel coordinates is that large data might cause clutter which makes interpretation more difficult.

2.6.5 Coordinated, Multiple Views Visualization

Mondrian is a general purpose information visualization system. It allows multiple displays to represent one data set and links them by brushing and selection [The02, MS08]. Figure 2.8 shows 5 coordinated views using bar charts, parallel coordinates and scatterplot matrix, on our five-variate data: task, document, page number, document feature and viewing time. The viewing time on documents, pages and features in the three bar charts is sorted in ascending order. Picture (Pi) and figure (Fi) have nearly the equal importance. The Page/Time bar chart reveals that participants focus on the first few pages, and quickly skip over the last pages. As we brush heading (He) from parallel coordinates, the other views are updated. But the multiple views can only deal with a single table at one time. If we need to compare the participants'

subjective scores and their estimated viewing time on the document features, we have to work in parallel with tables describing the pre-questionnaire.

The power of the Mondrain is its ability to visualize arbitrary dimensions of a data set separately. Due to the limitations of screen resolution, multiple views in Mondrian may be difficult to display and interact on large data sets simultaneously. Also, it can be difficult to infer which combination of visualizations is suitable and sufficient for HCI researchers to analyze their experimental data and solve the queries.

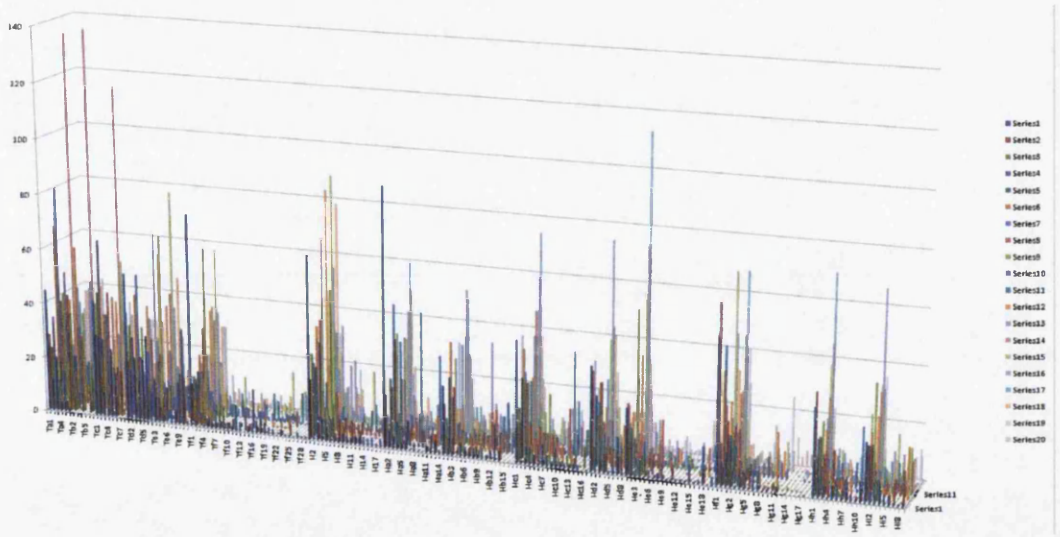


Figure 2.9: This figure display an overview of all 20 participants' status during the experiment in EXCEL 2007 [Mic07]. The X-axis is mapped to the document, Y-axis to the participants and Z-axis is the time spent on viewing documents. This visualization provides an interesting overview of the data.

2.6.6 3D Bar Chart Visualization

There are 11 distinct visualization techniques in Microsoft Excel, including various 3D visualizations for general use, and each of them has multiple variations. HCI researchers may use Excel to organize their raw data, in which arbitrary table columns can be easily mapped to the visual attributes. As shown in Figure 2.9, this provides an interesting bird's-eye overview of all the participants reading behaviors on documents and pages in the experiment. From this visualization, each individual participant's reading pattern can be displayed. But it suffers from occlusion problems. This might be addressed by the user interactions, such as selection and smooth zooming, rotation, and panning. But these dynamic manipulations and user navigations are not supported in Excel. Although there's a lot of debate on 3D interface [Shn03, TC09], considering our data set is semantically rich which contains documents, pages, participants and viewing time, we believe that the 3D bar chart is a way to further explore the individ-

ual participant's reading pattern provided that the software is able to offer enough interaction support.

	Interaction						Data Types					
	Filtering											
Tools	Browsing	Querying	Linking	Brushing	Dynamic Projection	Dimension Manipulation	1-,2-,3-D	Hyper Dimension	Tree	Text	Network	3D Viz
ManyEyes	√				√	√	√		√	√	√	
XMDV	√	√		√	√	√		√				
Mondrian	√		√	√	√	√	√	√				
Treemap 4.1	√	√				√			√			
TopCat	√	√					√					√
Office 2007	√						√					√

Figure 2.10: For each tool, we summarize its interaction techniques and supported data types. In addition, whether a tool contains 3D visualizations is also recorded. In every cell of the table, tick denotes the specific interaction or data type is supported in that tool, white space denotes such interaction or data type is not supported.

2.7 A Brief Subjective Rating of Tool Usability

Our goal in this chapter is not a general comparison of information visualization tools, but rather a specialized comparison dedicated solely to the investigation of document triage data. The beneficial visualization tools presented in this chapter are XMDV [War94a], Mondrian [MS08, The02], ManyEyes [VWvH+07] , TreeMap 4.1 [Kob04], TopCat [M. 05] and Microsoft Excel 2007 [Mic07]. They provide a variety of visualizations and integrate with different interaction options. These interaction designs of each tool are systematically applied to every visualization component within that tool. According to the taxonomy of Shneiderman [Shn96] and Keim [Kei02], data types to be visualized can be categorized as 1-, 2-, 3-dimensional (color is mostly used to depict the third dimension in most of the tools), hyper-dimensional, text, tree and network data. In addition, based on Keim [Kei02] and Kosara [KHG03b]'s work and the need for visual exploration on document triage data, the most frequently used interaction techniques include filtering, brushing, linking, dimension manipulation (the dimension manipulation includes dimension reduction and re-ordering options) and dynamic projection. The filtering can be achieved by either a direct selection of desired subset (browsing) or by a specification of properties of the desired subsets (querying) [Kei02]. The dynamic projection refers to dynamically change the projection of multi-dimensional data, such as Matrix Chart in ManyEyes, and Scatterplot Matrix in XMDV and Mondrian. In this section, we present a brief summary for the tools introduced in this chapter. Our summary is based on the tools' interac-

tion designs and the scalability to various data types, as shown in Figure 2.10. ManyEyes is able to handle all those listed data types except for high dimensional data. During the document triage study, data gathered would usually be from an excel spreadsheet, XML document or a text file. Manyeyes provides a good precedent to build upon regarding raw data input for custom visualizations. Since ManyEyes is deployed on the web, it saves a lot of time for the user during software installation and configuration compared with other desktop applications. To use this application, all we need is a username and password. In terms of ease of use, the ManyEyes is no doubt the best out of the six tools to our investigation. However, because of the social and collaborative nature, the data uploaded in ManyEyes will become visible to the public. This limits its usage with respect to data privacy.

XMDV and Mondrian, as complements to ManyEyes, are proficient in visualizing high dimensional data. XMDV features interactive, proximity-based clustering, which is effective for reducing the clutter caused by large data sets. But structure based bushing can be complicated to use for HCI researchers.

Mondrian offers coordinated multiple views (CMV) which effectively unveil different facets of the data. Compared with XMDV, it provides a greater choice of visualizations. As well as high-dimensional data, Mondrian is also quite effective in plotting large, low-dimensional data. The input data format in Mondrian is also more flexible. However, except by changing the alpha value, Mondrian does not provide more advanced clutter reduction techniques, such as the clustering offered by XMDV.

TreeMap 4.1 is specifically designed to implement treemaps. Compared with ManyEyes, it offers much more interaction options, such as numerical aggregation, various layouts, filtering and etc. However, with respect to the aesthetic feel of the visualization, the HCI experts prefer ManyEyes which provides more aesthetically pleasing color map and smooth animation when traversing through the different hierarchies.

Excel and TopCat are the only tools offering 3D visualization through the tools presented in this chapter. Although there is a lot of debate on 3D visualization [Shn03], it's surprising that the HCI researchers show more preference in 3D scatterplot and bar chart shown in Figures 2.3 and 2.9. However, in order to completely exploit the potential of 3D visualizations, interaction supports are very important. Although changing the viewing perspective, such as rotation, zooming and pan, are provided in Excel and TopCat, shading, which can effectively depict the depth information, is missing in both tools.

The first factor that became apparent is that no one visualization or tool on its own can identify all patterns and behaviours needed to be tested. Also, from the Table 9, we can see that no tool is able to support all of the data types and interactions. Furthermore, some visualizations such as bubble charts can cause the researcher to miss patterns and make false inferences, such as introduced in Section 2.8. In the light of this, it would be reasonable for a bespoke tool to include several visualizations in parallel. Therefore, a coordinated multiple view application allowing for a) several visualizations of the same data and b) one visualization with different data sets is needed. Ideally, visualizations for document triage data would include: line graph, 3D stack graph, treemaps and parallel coordinates. Overall, visualizations are underused in the HCI community as a means of interacting with extracted data sets. In this research we have explored the ways in which the visualizations enrich the exploration of relationships between different data sets of the same study. As an exploratory tool, using these

visualizations provides insight into hypotheses formulation about our data that is not evident from raw material. It is the aim of future work to apply the visualizations presented here, as well as further visualizations to not only explore the deciphering of the raw data, but to also assist users performing triage in making inferences about their material.

2.8 Domain Expert Review

The domain experts were impressed to see a multitude of visualizations that can represent their data. What became immediately evident was the ease and speed at which these visualizations could be produced. They systematically went through the visualizations identifying the immediate inferences that would have been possible before statistical analysis, but also factors that may obfuscate useful hypotheses from being formed. Analysis thus far has relied on statistical scrutiny such as t-tests. Although these are necessary for verifying a relationship or pattern they do not provide good means for exploration of the data. Here, they discuss the most significant observations and compare some of the visualizations presented in the chapter.

In general, the visualizations produced were applicable to the specific research in the document triage process. However, due to the nature of the experiment, the visualizations produced could be easily adapted to suite any researcher in the digital library and HCI field dealing with reading and searching for information on documents.

There were many visualizations that allowed for a synopsis of the participants' behavior and time distribution. Such visualizations included the 2D and 3D stack graphs in Figure 2.3, parallel coordinates in Figure 2.7, treemap in Figure 2.5, and matrix chart in Figure 2.6.

A more representative but slightly cluttered visualization is that of the matrix chart in Figure 2.6. This, they noticed, might be deceiving in presented false importance representations. For example a feature which is viewed for a large amount of time may suggest three possibilities: high distribution of this feature, large viewing time on the pages containing this feature, or both. Therefore, to this respect the further careful analysis is needed to distinguish its importance and make some helpful hypothesis.

Closely related in information representation is the treemap visualization in Figure 2.5. Although the same drawbacks that were mentioned for the matrix chat could also be given for the treemaps, the flexibility that this visualization offers in manually changing the hierarchy of the data to be processed gives it an advantage. Furthermore, the representation areas give a much clearer means of comparing features and timings.

Another beneficial representation of the data is found on the 3D stack graph in Figure 2.3. Beyond giving more information that the closely related 2D stack graph (which basically gives them the average values of all the pages) it allows us to detect further interesting behaviours worth exploring. For example, they notice the importance of the first page, but also the steady decline in attention as the page count increases. They can also detect the peaks close to the ends at the end which requires scrutiny, but also that the decline in attention is mostly steady. The 'anomalies' in the decline attest to one of two things: a) a sharp drop in attention on a specific point in the document or b) an increased amount of attention. They can therefore infer that further features also attract attention and test for the impact each feature has on attention.

One of the most interesting visualizations they came across was that of parallel coordinates in Figure 2.7. The features of a document and the influence they have time wise on participants constitutes a very important part of our data pool. This visualization gives them a clear image as to the percentage of time spent on those features in an-easy-to-compare format. Although there is great potential for this specific visualization, they do have two criticisms. The first is with regard to the upper and lower limit of the vertical axis. For every feature the maximum percentage time is set to the upper limit and therefore giving a false comparison between the feature values. This should be remedied in order to facilitate clearer comparative abilities. Another improvement which would increase comparative ability between data sets would be to be able to produce superimposed average values, standard deviations and multiple side by side visualizations of data sets.

Chapter 3

Angular Histograms: Frequency-Based Visualizations for Large, High Dimensional Data

Contents

This chapter is based on a publication from Geng et al [GLC+11]. Parallel coordinates, introduced by Inselberg and Dimsdale [Ins09, ID90b], is a widely used visualization technique for exploring large, multi-dimensional data sets. It is powerful in revealing a wide range of data characteristics such as different data distributions and functional dependencies [KK96]. However, one of the limitations with parallel coordinates is the clutter problem caused by rendering more polylines than available pixels. Overlapped lines often obscure the underlying patterns of the data, especially in areas with high data density.

Ben Shneiderman [Shn96] proposed the visual information seeking mantra: overview first, zoom and filter and details on demand, as visual design guidelines for interactive information visualization applications. However, this knowledge discovery process is hampered when rendering large data sets, because large data sets often cause a cluttered visualization which makes it difficult for a user to understand an overview of the data. If the user is unable to get a clear overview, it may become infeasible for them to determine which parts of the data can be filtered or zoomed in for more detail. In addition, a large data set slows interaction, making the data exploration process laborious. Therefore it is important to efficiently generate an information-rich overview of large data sets and enable a fast interaction process for the user.

A straightforward solution is to reduce the number of items to be displayed and present an abstraction of the data set. For the visual analysis to remain accurate, the graphical aggregation must preserve the significant features present in the original data. Up until now, there are many frequency-based approaches proposed for clutter reduction in parallel coordinates with histograms as one of the most widely used methods [BBP08b, KBH04, NH06, Wil96]. Histograms are able to depict the data distribution through a binning process, however, the traditional histogram only presents univariate data. For example, a single histogram can either represent the frequency of the data plots along every vertical axis [HLD02b, Wil96] or the angle of line-segments between pair of axes [DK10b], but not both at the same time.

In this chapter, we present angular histograms and attribute curves. These techniques consider each polyline-axis intersection as a vector. We visualize both the magnitude and direction of these vectors to demonstrate the principle trends of the data. Users can dynamically interact with the plot to investigate and explore additional patterns. We evaluate our methods on real-world animal tracking data sets and perform a comparison with the traditional alpha blending [Weg90a, WL97] and line-based binning algorithms [NH06].

The rest of the chapter is organized as follows: In Section 3.1, we review the previous work on clutter reduction in parallel coordinates. In Section 3.2, we present the algorithms for angular histograms and attribute curves. In Section 3.3, we demonstrate interaction design including angular filtering, selection and brushing. In Section 3.4, we present some use cases with respect to cluster analysis, linear correlation detection and outliers analysis. In Section 3.5, we discuss the performance of our visualizations.

3.1 Related Work

As a compact visual representation the parallel coordinate plot displays an n-dimensional data tuple as one polyline that intersects the parallel axes of each data dimension. Similar to other information visualization methods [dOL03, ED07, KK96, UTH06, WB94], the parallel coordinate plot suffers from overplotting which causes a cluttered visual representation. This is further hindered by the quantity of data points that are being plotted in a limited screen space. This drawback hampers further data analysis, such as investigating correlation and clusters.

In this section, we concentrate on previous work on parallel coordinates for large data sets. Generally, the clutter reduction methods for large data sets can be categorized as: alpha-blending, clustering, focus+context and frequency and density plots. We provide a brief overview of the literature on these methods.

Alpha Blending: Edward J. Wegman [Weg90a, WL97] represented the density of plots with transparency. In his method the sparse parts of the dataset fade away while the more dense areas are emphasised. This works well with small datasets, however, with large datasets the range of the data is much greater and consequently it is more difficult to fully represent the fidelity of complex datasets. It is difficult to obtain a clear understanding of patterns and clusters, and outliers may get lost.

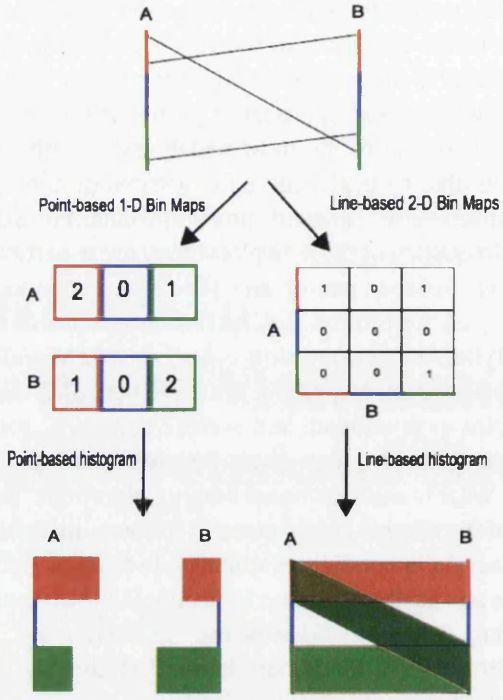


Figure 3.1: This figure shows (top) the original parallel coordinates. For each axis, three uniform bins or intervals are divided and depicted by different colors (red, blue and green); (middle) the two types of bin maps, with data frequency represented by the value displayed in each bin; (bottom) the two types of the histograms. For the point-based histogram on the left, the data frequency is mapped to the length of histogram bar. However, for the line-based histogram on the right, the frequency information is depicted by the alpha value of the histogram [NH06].

Clustering: Fua et al. define large data sets as containing 10^6 - 10^9 data elements or more [FWR99]. They adopt Birch's hierarchical clustering algorithm [ZRL96], which builds a tree of nested clusters of lines based on proximity information. Proximity-based coloring was introduced to demonstrate clusters, and transparency to show the mean and the extent of each cluster. Then multi-resolution views of the data can be rendered. In addition to hierarchical clustering, partitioning clustering, such as the K-means algorithm is also widely used [Mac67]. Johansson et al. [JLJC05] transform each K-means-derived cluster into three high precision textures, namely an animation, outliers and structure texture, and combine them into a polygon. Transfer functions are provided to highlight different aspects of the clusters.

Focus+Context: Wong et al. [WB96a] develop a multi-resolution display using wavelet approximations, where the brushed data is displayed at a higher resolution than the non-brushed data. Ellis et al. propose a focus+context viewing by the use of auto sampling and a sam-

pling lens on parallel coordinates [ED06]. They investigate three ways to calculate the degree of occlusion from overlapping polylines, and describe a raster algorithm as the most efficient metric. The sampling rate can then be automatically determined by the measure of clutter. Novotny and Hauswer develop another focus+context visualization using binned parallel coordinates [NH06]. Binned parallel coordinates are used for context views and traditional polyline-based parallel coordinates are used for focus views. However, for binned parallel coordinates, the uniform, equal-sized histogram bins do not allow for finer-resolution views of the data. Ruebel et al. [RK08] extend Novotny and Hauser's work, and propose adaptive histogram bins which use the higher resolution in areas with high data density. Their adaptive binning is able to represent general data trends more accurately.

Frequency and density plots: One of the ways to reduce clutter in parallel coordinates is based on data frequency. With this approach, the data is often aggregated and filtered by a binning process [AdOL04, BBP08b, Car91, NH06, RTT03]. In general, binning is the process of computing the number of values falling in a given interval or bin and storing them in a bin map. The data frequency can then be visually represented by a histogram. In parallel coordinates, the bin map can either be line-segment based which stores the frequency of the line segments connecting the adjacent axes, or point based which stores the frequency of the data points along each axis, as shown in Figure 3.1.

Much previous work adopts bin maps which yields line-based histograms [BBP08b, NH06, RK08]. They are effective at revealing clusters and outliers while further interaction support is needed to help the user select and brush interesting data and explore useful information. We find that one-dimensional point-based histogram is effective in revealing an overview of the data [HLD02b, Wil96], but such a histogram fails to depict the relations between the data axes. In this chapter, we extend the point-based histogram to a vector-based approach. We use histograms as the visual aggregation of both the frequency and the direction of the polyline-axis intersections. It offers the user an information-rich overview of the data. By introducing angular information from the polyline-axis intersections, our angular histograms and attribute curves are able to depict the relationship across the data attributes. The user is able to interact with the visualization through brushing and filtering, to further explore and analyse the data. We compare our result with the line-based [NH06] histogram.

3.2 Fundamentals

Our angular histogram and attribute curves are based on a vector-based binning approach. Through their utilization they provide the user with a rich overview of the underlying data and a better understanding of the data that cannot be gained from a traditional point-based histogram view. The use of the vector-based binning approach affords several advantages: First, it requires lower space complexity ($O(n)$) compared with the line-based approach ($O(n^2)$), where n represents the number of bins divided on each axis [NH06]. Second, it reveals the relationship of the plots between neighboring axes. Third, users can interact with the visualization by selection and brushing for further visual analysis.

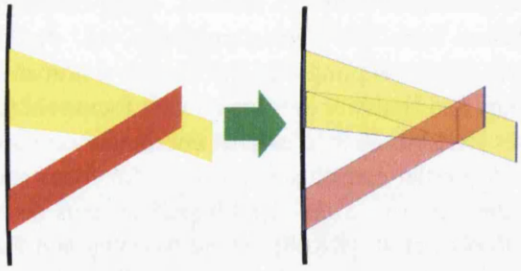


Figure 3.2: This figure shows two downward and one upward histogram bars overlapped. Alpha blending is applied to make the histogram bars visible in different layers. The silhouettes of each histogram bar are also rendered.

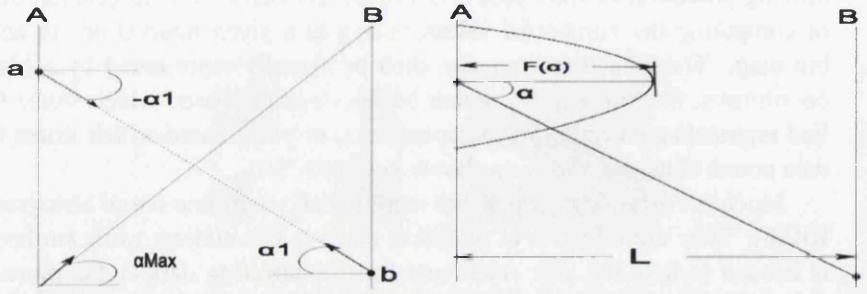


Figure 3.3: The left hand figure shows two attributes in parallel coordinates. A line segment connects a with b. The line segments of these data points map to unit vectors. We represent the unit vectors by the symbols **a** and **b**. We define the direction of the vector **a** as the angle between **ab** and the horizontal line starting from point a. Then α_{MAX}, which is the angle of a line segment connecting the opposite polar points of the two axes, is the maximal angle found between two axes. The right hand figure shows the mechanism of attribute curves. Curves starting at each data axis are pulled horizontally toward their neighbouring axis by the angular-frequency distance.

In this section, we use real world animal tracking data for some of our demonstrations [GJL+09]. Biologists at Swansea university have collected large amounts of data relating to animal movement by attaching sensors to individual subjects. The data here was captured at 8Hz for 8 hours and 40 minutes. In this chapter, we select 10 important data attributes which result in 1,048,566 records. The data attributes include: two accelerometers attached on the animal recording the acceleration parameters in X, Y and Z directions and an environment sensor recording the temperature, light-intensity (Infra depth) and pressure from the outside environment. This data set can be plotted using traditional parallel coordinates, but suffers from heavy overplotting, as shown in the top of Figure 3.4.

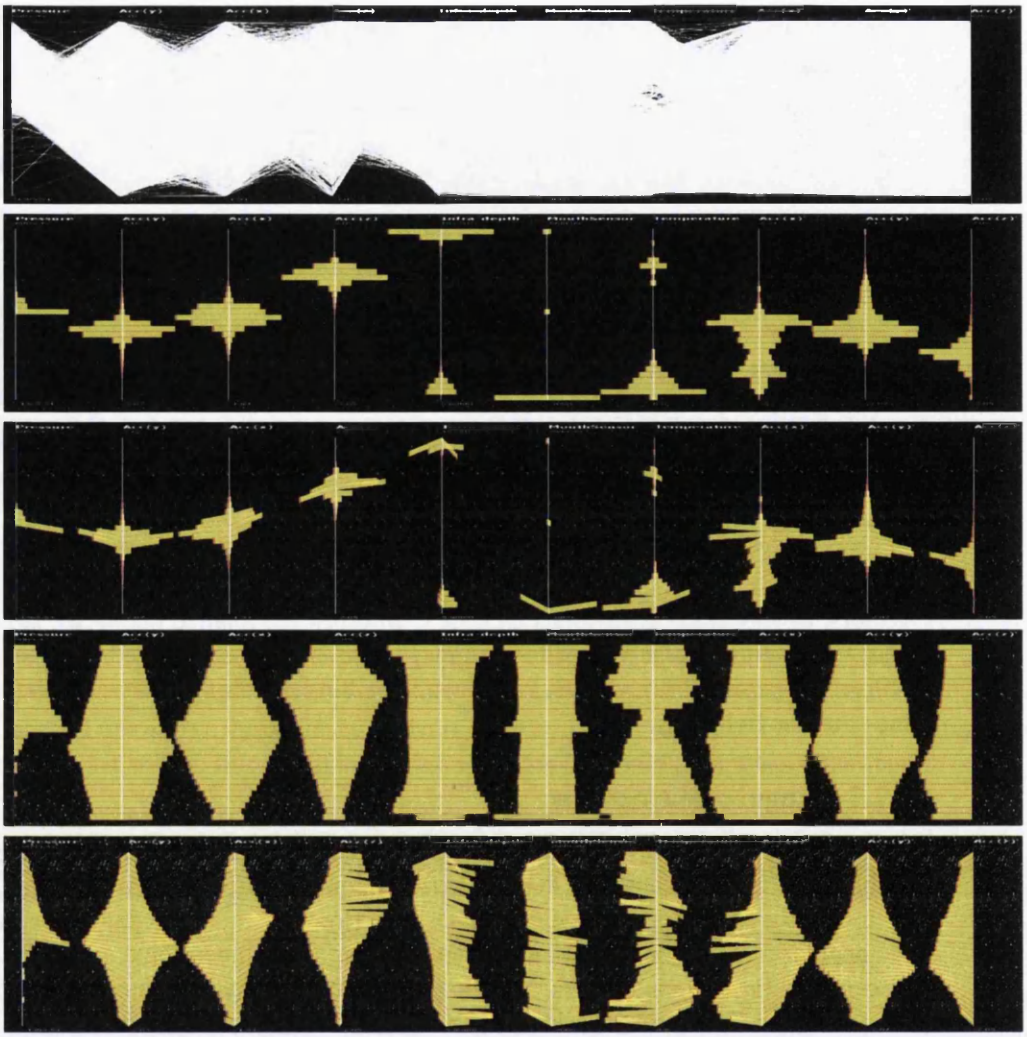


Figure 3.4: This figure shows the original parallel coordinates on animal tracking data (1st row); standard histogram overlay (2nd row); angular histogram overlay (3rd row); logarithmic histogram overlay (4th row) and logarithmic angular histogram overlay (5th row).

3.2.1 Vector-Based Binning

The standard histogram is widely used for estimating data frequency and density. It classifies the data into uniform, equal-sized intervals. Each bin is assigned an occupancy value according to the number of data items belonging to it. From the perspective of visualization, the histogram is a visual abstraction that aggregates the univariate data, where the height of the histogram bar is mapped to only one variable or feature. However, when displaying only one of these features it is hard to represent a complete overview of the data. If we map the slope of each line segment to a direction, then the polyline segment-axis intersections can be treated as unit

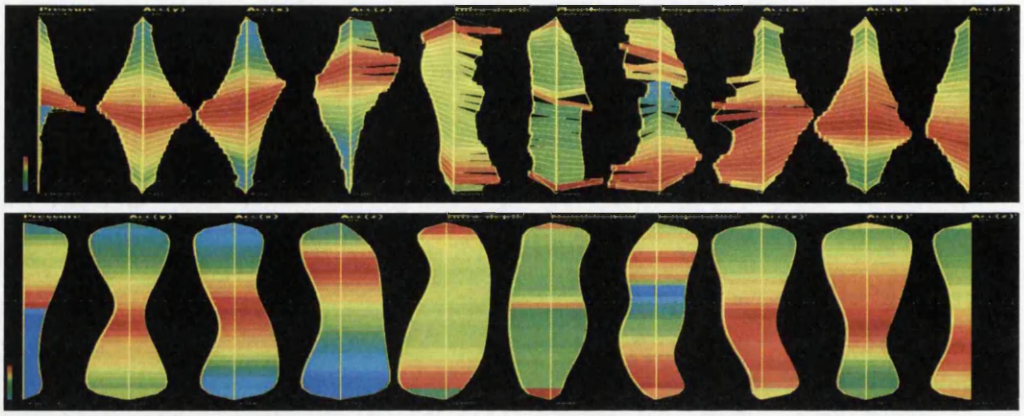


Figure 3.5: This figure shows the angular histogram and the attribute curves of the animal tracking data set. Color is mapped to the data density. Red indicates the largest frequency and light blue the smallest.

vectors, as shown on the left of Figure 3.3. In order to visualize the vector aggregations, at least two features, namely direction and magnitude, have to be encoded at the same time. We utilize two parallel bins on each vector with one bin recording the direction information and the other the frequency.

3.2.2 Angular Histograms

The standard point-based histograms are initially rendered in the second row of Figure 3.4. The height of each histogram bar is mapped to data frequency. From this visualization, we are able to discern the scalar distribution along each axis. However, this histogram representation lacks the angular information from polylines intersecting each dimensional axis. Thus we cannot discern or infer the relationships between the neighboring data attributes.

In Section 3.2.1, we introduced vector-based binning. The magnitude and direction of the vectors along each axis of the parallel coordinates are aggregated. Here we propose the angular histogram as an extension to the standard. The basic idea is that for each histogram we calculate the mean angle of the vectors and rotate the histogram bars by this angle. Then again the histogram bars can be considered as a vector, with length equal to the data frequency and the direction as the average angle of all its underlying polyline segment-axis intersections, as shown in the third row of Figure 3.4.

Different histogram bars on the same axis might overlap when rotated by a certain angle. We can apply alpha blending and silhouettes on the histogram bars to overcome the overlapping problem, as shown in Figure 3.2.

Although the angular histogram is able to convey the vector distribution, it still suffers from some drawbacks. The end points and the width of histogram bars determine the overall profile of the original, underlying line-segment distribution curve. When the bin width is too large it can cause undersmoothing, and when too small oversmoothing. A common statistical method for smoothing a data distribution, such as KDE (kernel density estimation),

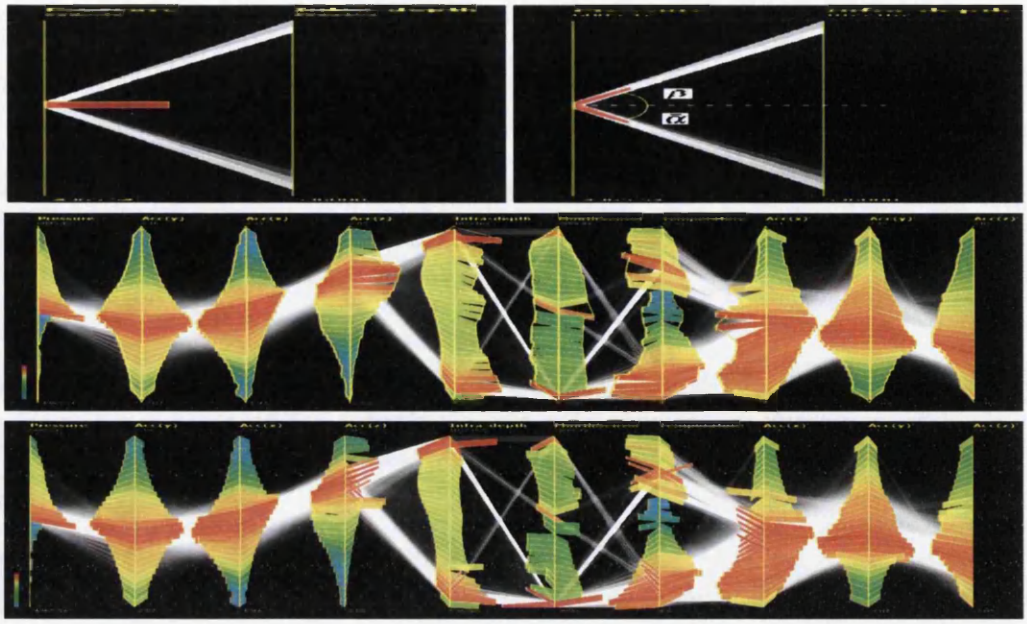


Figure 3.6: The first row shows an example angular histogram splitting is needed. The second row shows the original angular histogram using average angle of the animal tracking data set. The third row shows the divided angular histogram with $\xi = 0.2$ and $T = 80°$. For comparison purpose, we use the line-based histogram to render the underlying major data trend.

GMM (Gaussian mixture model) suffers from the computational complexity (particularly in high-dimension spaces) and the dependence on a bandwidth parameter or initial number of clusters [Sil86]. With this in mind, we have decided to leave the bin width as a user option. The user is able to interactively select the number of bins in each axis and obtain the corresponding angular histograms both locally and globally. Global bin selection applies the bin width to all histograms across all axes. Whereas local bin selection allows the user to adaptively select the bin size in different areas. For example, the areas with high data density might require a smaller bin width and thus more bins to depict finer detail.

Due to their frequency-based nature, histogram bins with relatively low density can be difficult to detect. One way to address this problem is to use a logarithmic histogram as shown in the fourth row of the Figure 3.4. The corresponding logarithmic angular histogram is rendered in the fifth row of Figure 3.4. From this visualization, low-frequency histogram bars and their directions are preserved.

When the histogram bars are rotated by a given angle, it's more difficult to discern and compare their relative lengths. It often happens that a given large data set is not balanced but is skewed. To address this problem, we can apply a color map on the histogram bars to represent the data density. In order to enable smooth transitions between the angular histogram bars, the frequency curve which connects the middle points of the boundaries of all histogram bars is rendered, as shown in the top of Figure 3.5.

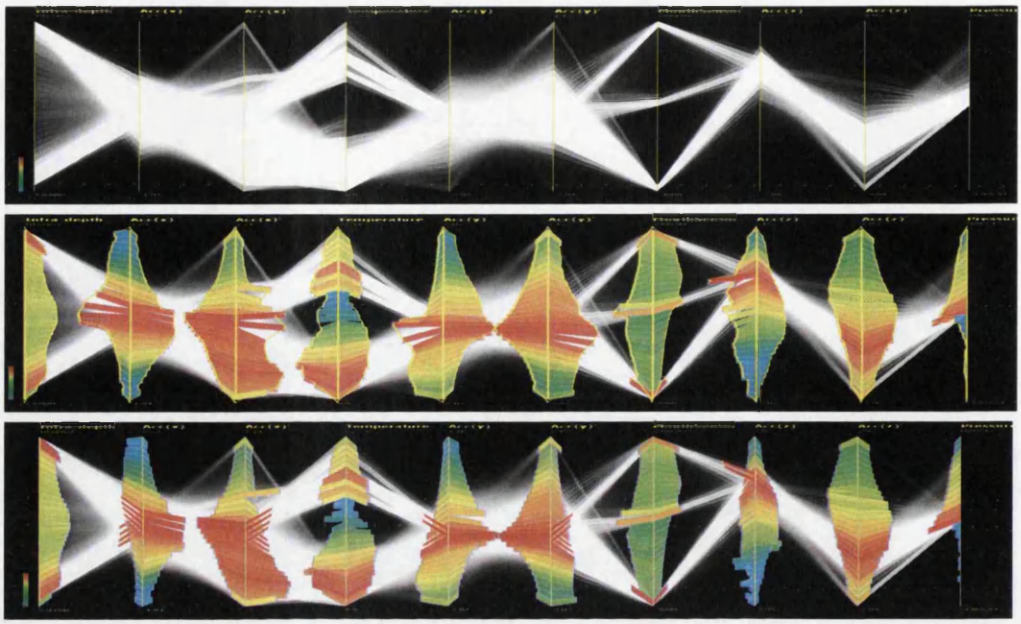


Figure 3.7: This figure shows the line-based clustering on a different ordering of our animal tracking data set (top); the angular histogram using average angle (middle); the divided angular histogram with $\xi = 0.2$ and $T = 80°$ (bottom).

3.2.3 Divided Angular Histogram

In the previous sections we introduced the angular histogram where the direction of each histogram bar is represented by the average angle. Although the mean value is a representation of the central tendency, it can be sensitive to extreme values (e.g., outliers) and the standard deviation might become significant. In order to accurately display the profile of the data trend, we propose the divided angular histogram as one of our user options.

On the left of the first row in Figure 3.6, we observe that there are two data trends passing through a histogram bin, pointing upward and downward respectively to the neighboring axis. If we calculate the average angle of these vectors, their positive and negative slopes cancel each other and lead to a flat histogram angle. In order to truthfully depict the data trend, we split the histogram bin into two separate groups: one contains vectors with an upward slope, such as b on the left of Figure 3.3. The other contains vectors with a downward slope, such as a in Figure 3.3. For each bin, we quantify the frequency of the upward and downward vectors, which can be denoted by n and m respectively. We also calculate the average angle for upward and downward vectors, which can be denoted by $\bar{\beta}$ and $\bar{\alpha}$, as shown in the right of the first row in Figure 3.6. In this figure we are able to see the original histogram bar is divided into two separate groups with one pointing upward and the other downward.

Because the splitting process increases the number of histogram bars displayed on the screen and might introduce clutter, we choose to split only a certain number of histogram bars to reveal the major data trend with more accuracy while avoiding the clutter problem. Two

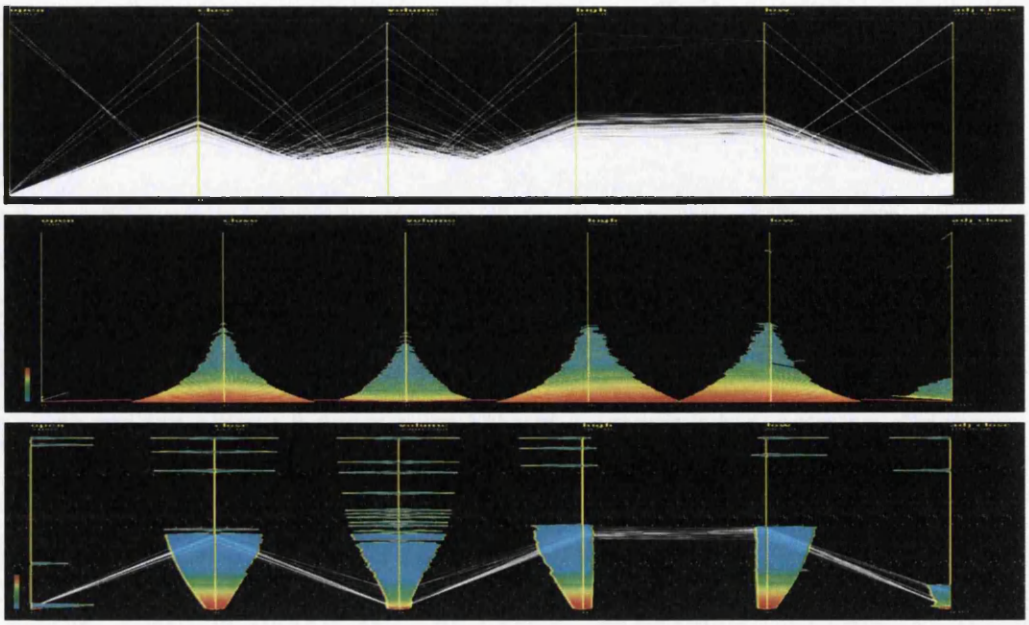


Figure 3.8: This data represents the daily volume of transactions, the opening price, the closing price, the highest and lowest volume of transactions in NASDAQ stock market from 1970 to 2010 [Inf11]. We see standard parallel coordinates (top); logarithmic angular histogram (middle) and attribute curves (bottom). The bin number is set to 100. The middle of the fourth axis is brushed and the underlying polylines are rendered.

user-centered approaches are provided to specify the number of histograms to be divided. The first approach enables the user to directly select and split any histogram bars they are interested in. The second approach defines a condition in order to automatically filter out undivided histograms. The condition can be expressed as: if the difference between the number of upward and downward vectors is small and the angle between the upward and downward vectors is large in a histogram bin, then this histogram bin is divided. This condition can be formulated as follows:

$$(0.5-\xi < \frac{n}{n+m} < 0.5+\xi) \quad \wedge \quad (|\bar{\alpha}|+|\bar{\beta}| > T) \tag{3.1}$$

where ξ represents a small value and is in the range $[0,0.5]$. In our case, we set ξ to 0.2 to ensure the number of upward and downward vectors are close. T represents a threshold value and is in the range $[0°, 180°]$. In our case, we set T to 80°. For the undivided histogram, we still use the average angle of all vectors contained in that bin to represent its direction.

To offset the effect on the average angle caused by a small number of outliers, we can use the trimmed mean [HK05] only if either the number of upward or downward vectors dominates in a histogram bin, otherwise the histogram angle remains unchanged. This can be defined as:

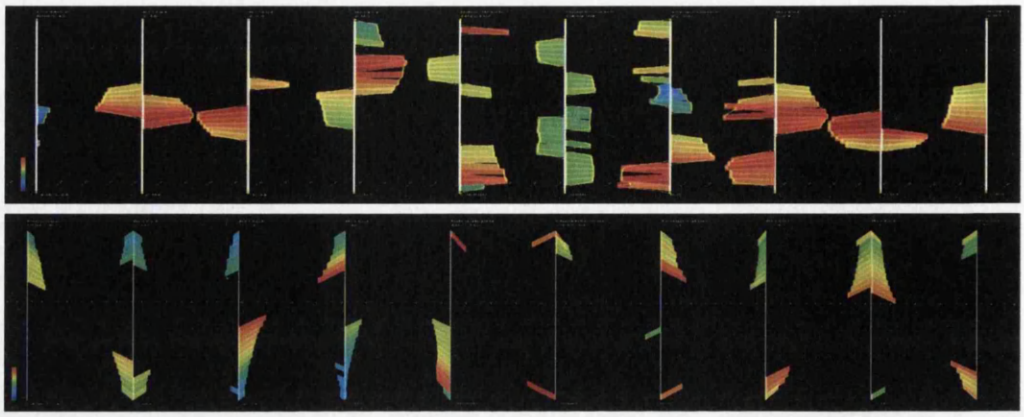


Figure 3.9: This figure shows two results of angular brushing on our histograms. The first row displays the angular histogram with flat angles and the second row depicts large angles.

$$\bar{\theta_{NEW}} = \begin{cases} \bar{\beta} & if \ \frac{n}{n+m} > 0.9 \\ \bar{\alpha} & if \ \frac{n}{n+m} < 0.1 \\ \bar{\theta_{OLD}} & otherwise \end{cases} \tag{3.2}$$

where $\bar{\theta_{NEW}}$ is the updated histogram angle and $\bar{\theta_{OLD}}$ is the original histogram angle. We choose a small threshold to avoid trimming too large a portion which may lose valuable information [HK05].

After histogram splitting and trimmed mean, we can obtain a more accurate cluster profile when following the direction of the histogram bars as shown in the second and third rows of Figure 3.6. The divided angular histogram works well in different orderings of our animal tracking data set, as shown in Figure 3.7. Besides providing a default setting for ξ and T, we also offer interaction support for the user to customize these parameters to explore the total solution space. The limitation of the divided angular histogram lies in its inability to split vectors with the same direction, such as a histogram bin that only contains upward or downward vectors. But we can further convey the standard deviation of histogram angles to the user, as discussed in Section 3.4.

3.2.4 Attribute Curves

Sometimes even a logarithmic histogram cannot depict outliers with very low frequency. The top of Figure 3.8 shows the daily total volume, open price, close price, the highest and lowest value of transaction in the NASDAQ stock market during 1970 and 2010 [Inf11]. Most data is gathered on the lower part of the axes and suffers from overplotting. The angular histograms shown in the middle of Figure 3.8 informs the user that no data exists in the upper half of the first four axes. However the original parallel coordinate plot demonstrates that there are few high values and volumes of transactions passing through the upper half of the axes. The reason that these values are not preserved in logarithmic angular histograms is because some low-

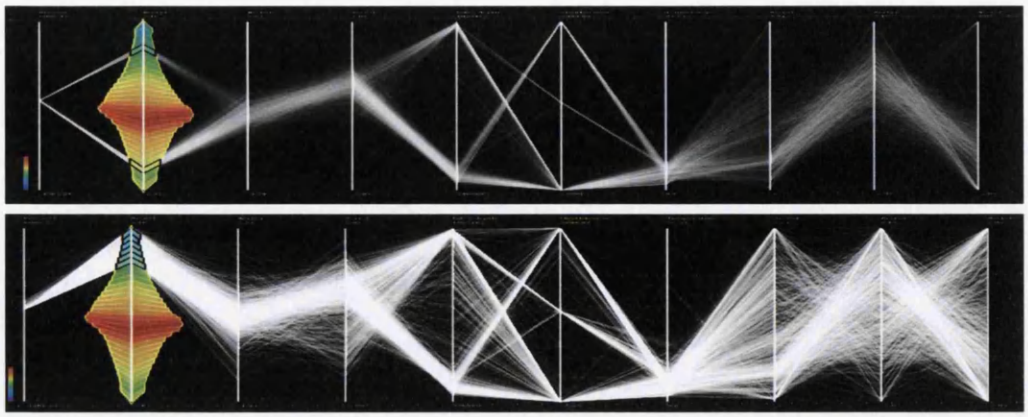


Figure 3.10: This figure shows the two ways of selection on our animal tracking data sets. The first row shows multiple selection. The second row illustrates the range selection.

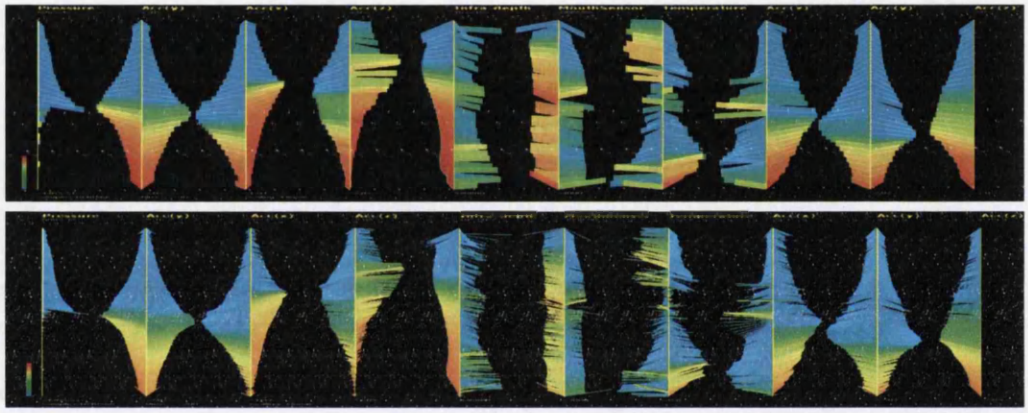


Figure 3.11: This figure shows two logarithmic angular histograms with (top) large, (bottom) small bin sizes on our animal tracking data set. The colour is mapped to the angle standard deviation. The larger deviations are represented using a red and the smaller using a light blue colour.

frequency histogram bins suffer from the low resolution of the display space and visually they look the same as the empty bins. In this section, we propose a user option called attribute curves to depict such outliers. In contrast to the discrete nature of angular histograms, attribute curves convey a smooth distribution of the underlying polyline data based on the angular frequency of the underlying polyline-axis intersections. Attribute curves are able to preserve the extreme values or outliers and indicate empty bins along the axis. In addition, they reveal the data distribution without the clutter associated with traditional parallel coordinate plots. In attribute curves the bin size that is used to compute the angular-frequency is the same as vector-based binning which stores vector direction in addition to frequency.

As shown on the right of Figure 3.3, curves starting at each data axis are pulled horizon-

tally toward their neighboring axis by the angular-frequency distance. The angular-frequency distance can be defined as follows:

$$F(\alpha) = \begin{cases} 0 & if\ k = 0 \\ \frac{d*(|\alpha|+\xi)}{(|\alpha_{MAX}|+\xi)} & otherwise \end{cases} \tag{3.3}$$

where $|\alpha|$ is the absolute average angle of the histogram bar and d is half the distance between the two adjacent axes. α_{MAX} is the maximal angle range of all histograms, as shown on the left of Figure 3.3. A small value ξ is added to the angle to make sure $F(\alpha)$ still has value when the histogram bar is horizontal ($\alpha = 0$). $F(\alpha)$ is zero if there is no data in the histogram bin (k refers to the bin frequency).

The larger the absolute histogram slope, the greater $F(\alpha)$ becomes. Then the slope distribution can be depicted as a smooth curve. The data density can be conveyed by luminance, where high density is mapped to more luminance and low density is mapped to less luminance. It is clear from the bottom of Figure 3.8 that the attribute curves preserve the few outliers on the top and indicate that these outliers have a large angle. Because we are using the absolute histogram angle in equation (3.3), our attribute curves will not suffer from the problem that the positive slope and negative slope may cancel each other, as discussed in Section 3.2.3. The absolute angle indicates the change rate of the data plot. A large angle often suggests a dramatic change of the data plot in a histogram bin from one axis to the next, while a small angle suggests a lack of change in the data plot between adjacent axes. For a better demonstration the middle part of the fourth axis in the bottom row of Figure 3.8 is brushed. We observe a steady transition of the data plots between the fourth and fifth axes and a relatively large change rate in the other adjacent axes from the brushed polylines. A few gaps in some part of the axes indicate the existence of empty histogram bins. From the color mapping, we are able to see a clear data distribution in the lower part of the axes which cannot be gained from the original parallel coordinates.

Attribute curves can also be applied to our animal tracking data sets, as shown in the bottom of Figure 3.5. By looking at the shape of the curve, we learn the relationship between the neighboring axes, and a principal data trend across attributes can be inferred from the density color coding. The user could also remove the parallel vertical axis from the attribute curves to form unparalleled coordinates. We recognize that attribute curves could pose challenges with respect to interpretation. This is true with many novel visualization techniques including parallel coordinates. This concern was not raised by the current users of our visualization techniques so far. However a full user-study is necessary to analyze interpretation. This is one of our future work directions. The algorithms we present are not intended to replace the traditional parallel coordinates visualization. They are meant as complements in order to facilitate exploration of large data sets.

3.3 Interaction

In the previous sections, we demonstrated how the angular histogram and attribute curves can be used for presentation and data overview. The next step is to allow the user drill down into

the interesting parts of the data and explore the useful information. To facilitate the information seeking mantra, we provide three types of interaction support including angular filtering, selection, and brushing on our attribute curves.

3.3.1 Angular Filtering

A angular filtering is similar to the work by Hauser et al. [HLD02b]. Angular brushing [HLD02b] is effective in revealing the relations between neighbouring data attributes by filtering the slopes of the line segments. However utilizing the angular brushing on the original large data sets may cause a cluttering problem and slows down the performance. Considering the angular histogram provides a visual aggregation of the vector directions, it can be filtered by the angles of the histogram bars. The user is able to define the range of histogram angles they are interested in and the histogram bars with angles in this range will be rendered. We demonstrate two brushing results in this section: one displays the angular histograms only with small angles, as shown in the top of Figure 3.9; the other displays the angular histograms with sharp angles, as shown in the bottom of Figure 3.9. The obliqueness of the angular histograms depicts the relationship of the data between the neighboring axes.

3.3.2 Selection

The user is able to select the histogram bars of interest and the original underlying polylines passing through the given bin will be rendered. The histogram bars can be selected in three different ways, including the singular selection, multiple selections and range selection. The singular selection enables the user to select any individual histogram bar. Multiple selections allow the user to select various histogram bars at the same time, as shown in the first row of Figure 3.10. The range selection allows the user select a range of histogram bars, such as in the upper region of the axis, as shown in the second row of Figure 3.10. We enable alpha-blending on the selected polylines to avoid clutter. In order to show the context, the angular histogram of the selected axis is also rendered.

3.3.3 Composite Brushing

The user is also able to select angular histogram bars from different axes and perform a composite brushing, such as an AND-brush or OR-brush [HLD02b]. In the context of large data sets we find that the AND-brush is particularly useful in reducing clutter and finding the major trend in the data. As shown in Figure 3.14, the angular histogram provides a context view while the focused polylines are rendered.

3.4 Use Cases

For demonstration, we used two different axis orderings of our animal tracking data set which cause serious clutter. In the following sections, we demonstrate our methods with respect to the performance of cluster analysis, correlation detection and outlier analysis. We compare our

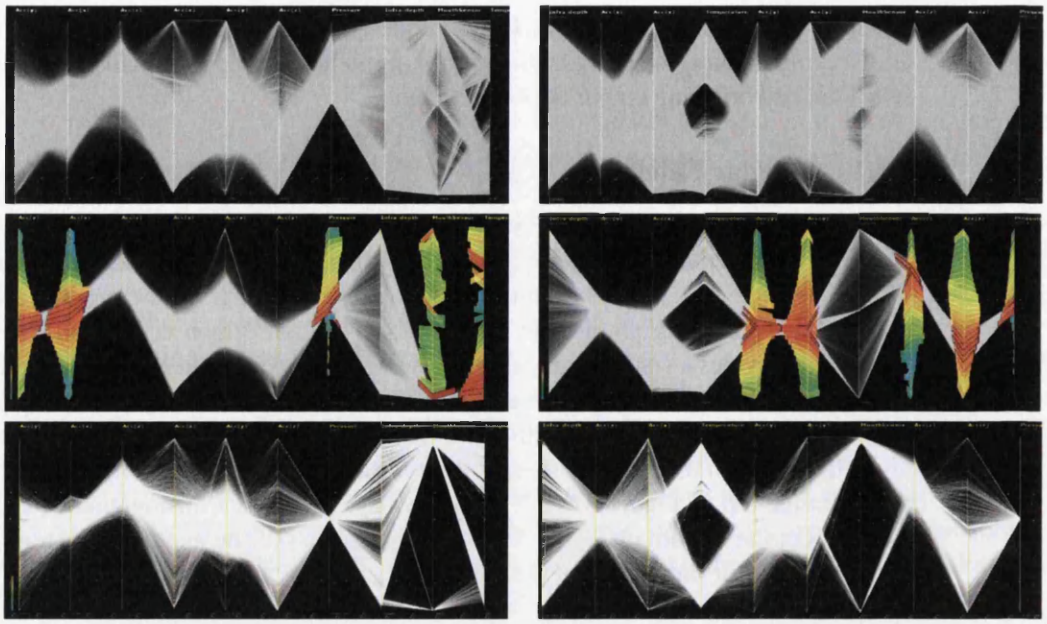


Figure 3.12: This figure shows the two axis orderings from our animal tracking data set. The first row shows the parallel coordinates rendered with a low alpha value. The second row shows the brushed major data trend, only the selected angular histograms are rendered to preserve the context view. Also, the selected histogram bars are rendered in black halos. The third row shows a complete cluster profile using both the AND-brush and OR-brush.

methods with alpha blending [Weg90a, WL97] and line-based binning [NH06]. Our techniques rely on the user to provide numerical orderings for nominal data.

3.4.1 Cluster Analysis

In the previous sections we introduced the angular histogram which indicates the path where the majority of data flows across the parallel coordinates. Although the divided angular histogram could reduce the loss of information, the standard deviation is introduced. In order to provide an accurate data overview we need to quantify and indicate such deviation to the user. The standard deviation of the angle of each histogram bar can be represented as:

$$\alpha_\varepsilon = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.4}$$

where n is the bin count, x_i is the angle of each vector and $\bar{x}$ is the mean angle of this bin.

We facilitate the user to utilize various color scales to depict this deviation information. Although some of the directions of the histogram bars are not fully representative due to the deviations, we inform the user of how far the vector directions deviate from the mean angle by the color mapping. Larger deviation indicates a higher possibility that the actual data in this

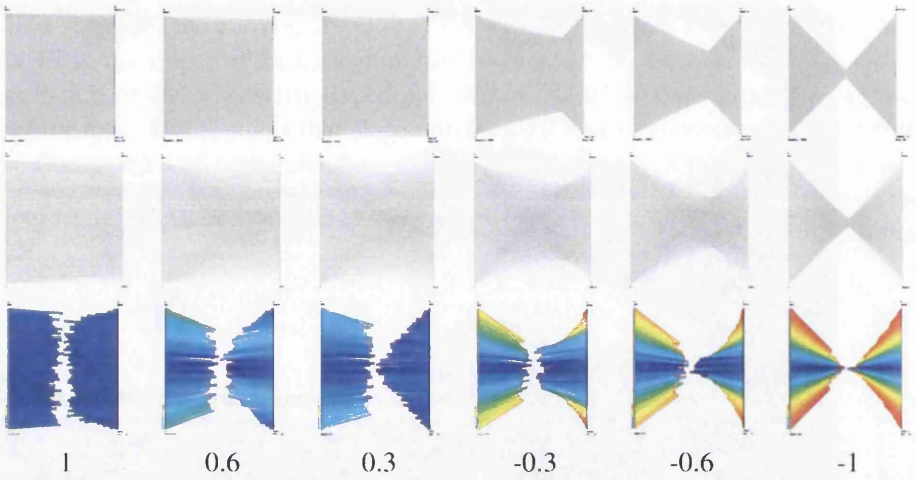


Figure 3.13: This figure shows a group of correlated sample data sets accompanied by the corresponding angular histograms and alpha-blending. The 1st row shows the original data set with 6500 rows. The 2nd row shows the same data sets but rendered in smaller alpha value (α = 0.002). The 3rd row shows the angular histogram of the correlated data set. The correlation levels (Pearson Coefficients [HK05]) from left to right are in descending order. We color code the sharpness of the angular histogram, from red to blue depicts the largest to smallest slope.

bin leads to different sub clusters. Sometimes using a smaller bin size may help reduce the standard deviations, as shown in Figure 3.11.

Since the angular histogram can represent an appropriate profile of the principle trends, the user is immediately drawn to the sets of interest. The user can then select the high density histogram bars and render the original polylines passing through them by the AND-brush, as shown on the second row of Figure 3.12. We compare the alpha blending with our brushed polylines as shown on the top row of Figure 3.12. From this comparison, we are able to see that the angular histogram is able to present various sub-clusters of the data. By the combination of AND-brush and OR-brush, the user is able to obtain a complete cluster profile which gives a clearer and more accurate overview of the principle data trend than the alpha blending, as shown in the third row of Figure 3.12.

3.4.2 Linear Correlation Detection

Jing Li et al. suggest that the original parallel coordinates are less effective than scatterplots for visual correlation analysis [LMvW10]. However, the conclusion drawn from that work was not based on a large data set. Millions of data points rendering on the screen causes serious clutter, which can make visual correlation analysis infeasible in both parallel coordinates and scatterplots. In this section we discuss how angular histograms can be used to enhance visual correlation analysis for large data.

The different levels of data correlation will impose various shapes on angular histograms. In order to observe the underlying rules for analyzing such shapes, we prepare a group of large

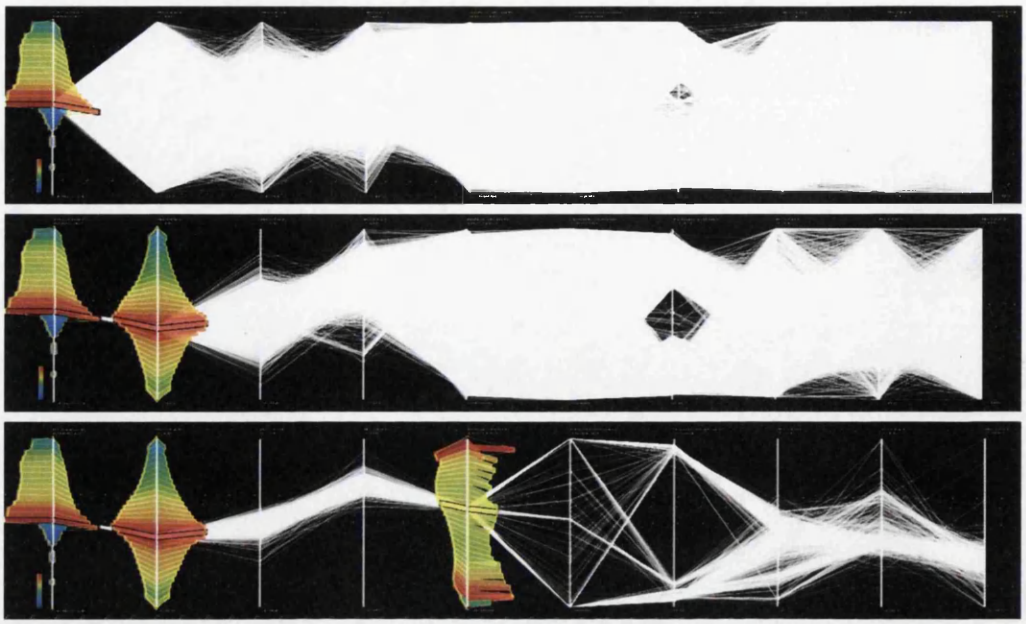


Figure 3.14: This figure shows an AND-Brush on our animal tracking data sets.

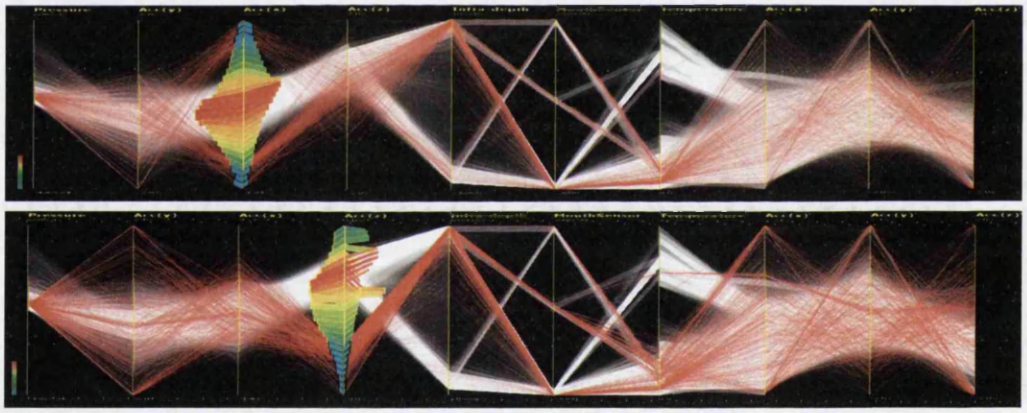


Figure 3.15: This figure shows two outlier-preserving visualizations. The underlying cluster is rendered by line-based histogram and the outliers can be brushed using the angular histogram.

sample data sets with different correlation levels for illustration. The sample data group is manually generated according to the work of Jing Li et al. [LMvW10]. We show six correlation levels whose Pearson Coefficients [HK05] range from -1 to 1, as shown in the first row of Figure 3.13. The second row of Figure 3.13 shows the same data but rendered with a smaller alpha value.

The third row of Figure 3.13 shows the angular histograms applied to the sample data sets. We color code the sharpness of the angular histogram, from red to blue depicting the largest

to smallest slope. As the correlation decreases, angular histograms reveal a series of changing patterns. First, the slopes of the histogram bars increase as the correlation decreases. Second, the distribution of the downward sloped points have a tendency to cluster around the upper region of the axis. Third, points that slope upward tend to cluster around the lower regions of the axis. Compared with alpha blending, angular histograms uncover the rate of change of the line slopes as the coefficient strength decreases: the closer the lines to the middle of the axis, the slower their slopes change and the closer to the lines to the polar ends of the axis, the faster their slope change.

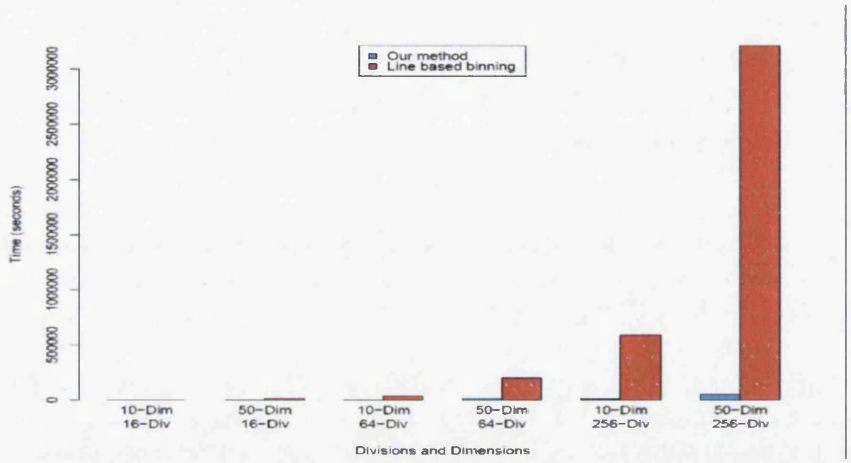


Figure 3.16: This figure shows the comparison of the number of bins constructed by line-based binning and our vector-based binning.

3.4.3 Outlier Analysis

In statistics, an outlier is an observation that is numerically distant from the rest of the data. How to present the major trend in the data while preserving the outliers is an important question in visualization design. In Section 3.2, we briefly discussed the strength of attribute curves using the NASDAQ data set. On the one hand the logarithmic histogram does not preserve very low density histogram bins, while on the other hand the attribute curves are able to visualize this and together the two techniques compliment each other. Thus they can be combined to enhance outlier analysis.

In this section, we specifically demonstrate how our angular histogram can facilitate outlier analysis for the traditional alpha-blending method. As shown in Figure 3.12, the top row is two different orderings of our animal tracking data set rendered in low alpha value. Although we set the opacity to a very low level, there are still some regions suffering from overplotting and the low frequency area is not preserved. From the eighth axis on the right image of the top row, we observe that the majority of the data is gathered on the middle part of the axis. If the user wants to know how the low and high values in this axis correlate with the neighboring axes then this is difficult to see with alpha-blending. Figure 3.7 shows the angular histogram

with the same data set and axis ordering as the one shown in the right column of Figure 3.12, we could see that the line-based histogram is unable to preserve the low-frequency area either. However, from the angular histogram shown in the third row of Figure 3.7, the directions of histogram bars give an indication of the relationship between the low-frequency data and the neighboring axes. The user could subsequently brush this area and render the polylines passing through these histogram bins.

Our angular histogram could also be incorporated with the line-based histogram, as shown in Figure 3.15. The underlying cluster can be rendered by the line-based histogram. From the angular histogram, the user is able to learn an informative overview of the data. Such overview will guide them to select and brush the interesting parts of the data, such as low-frequency bins, on top of the principle data trend, as shown in Figure 3.15.

3.5 Discussion

For a data set containing n dimensions and m records, with each of its attributes uniformly divided into k intervals, we need to construct $(n-1)k^2$ bins for storing the line frequency [NH06], whereas only $4nk$ bins are needed for the vector frequency. The comparison of the two methods is shown in the Figure 3.16.

Moreover, reordering the axis in parallel coordinates generates a new bin map. Blass et al. [BBP08b] discuss the way to handle reordering is to pre-compute the bin maps for all possible axis permutations. In this respect, our vector-based histogram has the advantage of taking up much less processing time and memory space than the line-based histogram, especially for data sets with high dimensions.

The line-based histogram [NH06] is able to present a clear representation of the principle trend in the data. However, due to the fact that it only aggregates the frequency of the lines between pair of axes, the data trend it reveals is not as coherent as our brushed polylines across all data attributes, as shown on the right column of Figure 3.12 and Figure 3.7. In order to enhance the clustering and outlier effect, the line-base histogram [NH06] utilizes different filters, such as Gaussian or Median filters. But except by changing the bin size, the user does not have much control over the visualization. Whereas in our vector-based histograms, we aim to facilitate the information seeking mantra for the user. The data overview is presented by the angular histogram. Following the direction of the high density histogram bars give a general cluster profile. The use of the logarithmic angular histogram and attribute curves depict the outliers in various levels to the user. When the user obtains a good overview, we offer various interaction supports for them to perform in-depth visual analysis. Our technique is not aiming at automatic cluster or outlier detection, instead we leave the control with the user by providing an informative overview accompanied with interaction. The user has a high degree of freedom in determining their interested parts of the data. We compared our angular histogram with the line-based approach in Figure 3.7 to ensure the overview of the data is accurate and not biased.

Chapter 4

Markov Chain Driven Multi-Dimensional Visual Pattern Analysis with Parallel Coordinates

Contents

This chapter is based on a publication from Geng et al [GWL12]. One of the promising algorithms for discovering principal data trends for large data sets in parallel coordinates is based on data frequency [AdOL04, BBP08b, Car91, NH06, RTT03], as discussed in Chapter 3. With these approaches, data is sometimes aggregated and filtered by means of binning. Due to the curse of dimensionality, most frequency-based approaches adopt a two-dimensional bin map which stores frequency of line segments between adjacent axes. A joint histogram is then rendered based on this bin map. A clustering or outlier detection method is limited to a two-dimensional subspace and the multi-dimensional features are not considered. Due to the dependencies inherent within multidimensional parallel coordinates, we need to discover and summarize the patterns which can propagate through n-dimensional space, as opposed to being limited to a two-dimensional subspace. In addition, high-dimensional data clustering heavily depends on the choice of subspace [KKZ09], which is not discussed in previous work.

In this chapter, we develop a probability model to guide the user to brush a subset of the data items which can represent major and minor trends in a data set. To achieve this, a weight value, which is determined by the joint probability of the n-dimensional data features, is assigned to

each polyline. A polyline with a higher probability value implies that a given data tuple is part of the principal trend, whereas a lower probability value implies there are few similar patterns present. In order to compute a joint probability value, we introduce the Markov Chain model [Rab89], which can be constructed and implemented using the binning method. Initially, each dimension of the original data set is divided into a number of uniform bins where each bin is interpreted as an individual state. Then every multidimensional data tuple can be represented by a sequence of states. A two-dimensional line-segment based bin map is constructed to compute a transition probability matrix. We provide various interaction techniques for the user, such as determining how much data they would like to brush for patterns and manipulating dimensions to define the Markov Chains. When rendering large data sets, sometimes the size of the underlying, brushed data trends can be large which might lead to overplotting and clutter. In addition to the traditional alpha blending, we also propose an improved angular histogram to represent and visualize the density of large data.

The contributions of this chapter are summarized as:

- We develop a Markov Chain-based probability model for multidimensional pattern discovery for large data sets with parallel coordinates.
- Our approach is able to preserve global data trends by composite brushing in higher dimensions without suffering from the curse of dimensionality.
- Our method is able to depict the coherent n-dimensional visual patterns better than previous line-based histogram representations.
- We propose an extended angular histogram to represent the data density when the brushed data sets are large.

We demonstrate our techniques on real world n-dimensional marine biology data in addition to several well-known data sets. The rest of this chapter is organized as follows: Section 4.1 discusses the previous work related to our approach. Section 4.2 demonstrates the key ideas to compute a joint probability value for a data sample using Markov chain. Once the probability values are obtained, they can be presented by a histogram or scatterplot, as discussed in Section 4.3. We also provide several interaction techniques for the user, such as a brushing method based on a probability value which can decompose the original data set into principal major and minor trends. Section 4.4 evaluates our proposed method based on a comparison with other well-known visualization techniques for large data sets. In addition, the user can create an arbitrary Markov chain by dimension reordering and manipulation. This is discussed in Section 4.5. In Section 4.6, we propose an extended angular histogram to depict the data density when brushing large data sets. Section 4.7 describes a case study with respect to a real world marine biology data set.

4.1 Related Work

One of the ways to represent a major data trend in parallel coordinates is based on data frequency. With this approach, the data is often aggregated and filtered by binning. In general,

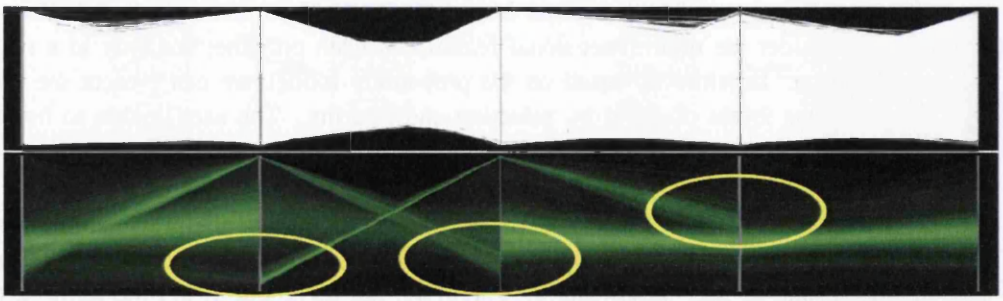


Figure 4.1: This figure shows the line-based histogram for the remote sensor data from the paper [NH06]. This method is able to capture the local data trend in the two-dimensional subspace, rather than global trend in higher dimensions. Highlighted in yellow are the discontinuous patterns due to the two-dimensional joint histogram representation.

binning is the process of computing the number of values falling in a given interval and storing them in a bin map. Data frequency can then be visually represented by a histogram.

In parallel coordinates, bin maps can either be line-segment based which store the frequency of the line segments connecting adjacent axes, or point based which store the frequency of data points along each axis [AdOL04, Car91]. Novotny and Hauser develop a focus+context visualization using binned parallel coordinates [NH06]. Binned parallel coordinates are used for context views, while the traditional polyline-based parallel coordinates are used for focus views. However, for the binned parallel coordinates, uniform, equal-sized histogram bins may not allow for finer-resolution views of the data. Ruebel et al. [RK08] extend Novotny and Hauser's work, and propose adaptive histogram bins which use a higher resolution in areas with high data density. Their adaptive binning is able to represent general data trends more accurately. Blaas et al. [BBP08b] optimize the data preprocessing for the binning method with respect to the data storage, histogram equalization and quantization. This facilitates fast exploration for large data sets. Muigg et al. [MHDG11] utilize a tensor representation to depict the direction of the data trend in dense areas of binned representation.

Because the bin map used in previous approaches is based on neighbouring dimensions, it inevitably introduces discontinuous patterns across multiple dimensions. What we obtain from these binning methods are the clusters and outliers in a one or two-dimensional subspace. We are unable to visualize n-dimensional data as a coherent feature, as shown in Figure 4.1. In Feng [FKLT10]'s work, this discontinuity between every two dimensions is regarded as uncertainty and Kernel Density Estimation (KDE) is used to enhance such information. However, the uncertainty they consider is also based on a two-dimensional subspace. A possible solution to address this problem is to build a truly n-dimensional bin map. However, as the number of dimensions increases, the total number of required bins grows exponentially, which is k^n for a data set with n dimensions and k bins. This can cause enormous memory demands even for a small number of intervals. In this chapter, we propose a novel approach to aggregate n-dimensional data tuples using a probability model based on a line-based binning method. A two-dimensional bin map between the neighboring axes is constructed to compute a transition

probability in our Markov Chain Model [Rab89]. The main contribution of our work is that we consider the multidimensional features of each polyline, not only in a two dimensional subspace. In addition, based on the probability model, we can present the principal trends at various levels of detail by selection and filtering. The user is able to fine-tune different parameters of the Markov model to obtain both major and minor data trends.

Clustering: The ultimate goal of our work is to improve the previous line-based histogram for multidimensional pattern discovery, it focuses on data extraction and filtering, rather than data grouping and classification. Therefore, our technique is not the same as the traditional clustering methods, such as hierarchical clustering [FWR99, ZRL96] or K-Means clustering [JLJC05]. A data item with very high probability suggests it is part of a data trend. If we are able to display all these high-probability data samples, then some trends or clusters can be visually discerned. In our approach a cluster or trend is implicitly revealed and we have no knowledge of its inherent structure. However, as noted by Wegman et al. [WL97], the separation between or among sets of data on any one axis represents a view of the data which isolates clusters. Therefore, although we are unable to represent the explicit cluster membership and structure, we can utilize color mapping on any given axis to visually separate the discovered multidimensional data patterns and trends. Traditional clustering is also prone to the problem of choosing appropriate seeds. A user may have no previous knowledge of the data sets thus it is difficult for them to choose appropriate seeding values and quantities.

Outlier Detection: Our method is not explicitly designed for outlier detection. However it can be used to detect and visualize outliers in higher dimensions using our joint probability distribution. The traditional density based outlier detection method, such as n-dimensional Kernel Density Estimation (KDE) which is based on a continuous density function, is computationally expensive especially when the size and dimension of a data set is high. Whereas in comparison our method with a complexity $O(nm)$ does not suffer from this problem, where n is the number of dimensions and m is the number of data items. Because a probability value generated by our method for each data sample indicates the number of similar patterns to it, a very low probability potentially suggests an outlier which is numerically distant from other data items.

4.2 Fundamentals

In this section, we will explain the key concepts behind our approach. Section 4.2.1 demonstrates the Markov Chain model developed for parallel coordinates. Based on this model, each multidimensional data tuple can be assigned a joint probability value. In order to quickly compute such a probability value, we pre-compute a transition probability matrix and store it in an external file, as discussed in Section 4.2.2.

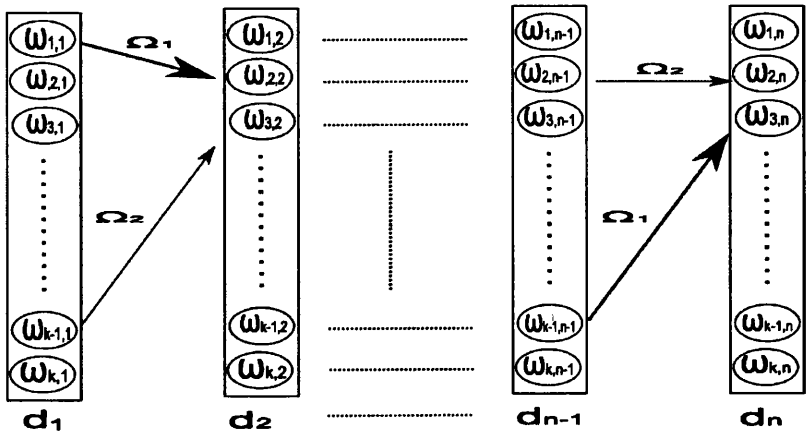


Figure 4.2: This figure shows the Markov Chain Model applied in parallel coordinates. Each vertical axis is treated as one time step and is divided into several bins or states. The thickness of the arrow for each transition, such as Ω_1 and Ω_2, depicts the joint probability value.

4.2.1 Markov Chain Model

In this section, we will explain the key concepts behind our approach. Given a data set $X = \{\mathbf{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,n})^T | 1 \leq i \leq m\}$ with m items of n dimensions, the binning method converts the original data into a frequency-based representation by dividing the data space into k multidimensional intervals, namely bins. The key idea behind our approach is to compute a joint probability value for each data item $\mathbf{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,n})^T$. Then we can introduce the Markov Chain to compute the joint probability of multidimensional data. A Markov Chain [Rab89] is a stochastic process that undergoes transitions from one state to another in a chainlike manner. The first-order Markov Chain defines that the current state depends only on the previous state and not on the entire past. In order to formulate the probability model, we firstly construct a one-dimensional binning for every data dimension. If the data in each attribute are divided into k intervals, then in total we will need kn bins for all attributes, which is denoted as $S = \{\Omega_i = (\omega_{1,i}, \omega_{2,i}, ..., \omega_{k,i})^T | 1 \leq i \leq n\}$, where $\omega_{i,j}$ represents the i^{th} bin in axis j. For every item $x_{i,j}$ in the data space, it can be converted to the bin membership by the function $\phi(x_{i,j})$, which returns the bin index that the data item $x_{i,j}$ belongs to.

To formulate our Markov Chain model, each bin $\omega_{i,j}$ can be interpreted as a state. A list of data dimensions, $D : d_1, d_2, ..., d_n$, can be treated as a temporal sequence. Every data tuple or polyline forms a sequence of transitions from one state to another over a series of time steps. Computing a probability of a data tuple $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,n})^T$ can be transformed to computing the probability of the list of states where this data tuple flows to, namely $\Phi_i = (\phi(x_{i,1}), \phi(x_{i,2}), ..., \phi(x_{i,n}))^T$, this Markov process is shown in Figure 4.2. A joint probability of each bin tuple Φ_i can be defined as:

$$P(\Phi_i) = P_0(\phi(x_{i,1}))\prod_{t=2}^{n} P(S_t = \phi(x_{i,t})|S_{t-1} = \phi(x_{i,t-1})) \tag{4.1}$$

where $P_0(\phi(x_{i,1}))$ is a stationary probability value, which can be represented by the data frequency within this bin.

In order to improve the numerical stability when using limited precision floating point numbers for product computation in equation (4.1), we take the logarithm of both sides. This can be defined as:

$$ln(P(\Phi_i)) = ln(P_0(\phi(x_{i,1}))) + \sum_{t=2}^{n} ln(P(S_t = \phi(x_{i,t})|S_{t-1} = \phi(x_{i,t-1}))) \tag{4.2}$$

Before defining the transition probability, we construct the two-dimensional binning which stores the line segment frequency between every pair of axes. Based on this bin map, we are able to build a $kn \times kn$ stochastic transition matrix, where k is the number of intervals and n is the number of dimensions as shown in Figure 4.3. Each element of this matrix can be defined as:

$$P(S_t = \omega_{i,j}|S_{t-1} = \omega_{u,v}) = \begin{cases} \frac{\|\omega_{i,j} \cap \omega_{u,v}\|}{\|\omega_{u,v}\|(n-1)} & if\ j \neq v \\ 0 & otherwise \end{cases} \tag{4.3}$$

where $\|\omega_{i,j} \cap \omega_{u,v}\|$ represents the number of common data items that both bins $\omega_{i,j}$ and $\omega_{u,v}$ share, i.e, the number of line segments joining the two bins between the axes j and v. $\|\omega_{u,v}\|$ is the number of data items in the bin $\omega_{u,v}$. n is the number of dimensions. If two states are from the same axis ($j = v$), then the transition probability is zero. In our model, we only consider the case ($j \neq v$) where the transition probability is the normalized conditional probability of state $\omega_{i,j}$ over state $\omega_{u,v}$.

Since a state transition probability matrix has to be a stochastic matrix, the sum of each row of the matrix has to be one. Based on equation (4.3), we can prove that our transition probability matrix satisfies $\sum_{v=1}^{kn} p_{u,v} = 1$, for all $1 \leq u \leq kn$, where $p_{u,v}$ represents a matrix element. We take the first row of the matrix in Figure 4.3 as an example. The transition probabilities in this row can be separated into two parts, one contains the states in the same dimension as $\omega_{1,1}$, the other contains the states in different dimensions:

$$\sum_{u=1}^{kn} p_{1,u} = \sum_{u=1}^{k} p_{1,u} + \sum_{u=k+1}^{kn} p_{1,u} \tag{4.4}$$

Based on equation (4.3), the first part of equation (4.4), namely $\sum_{u=1}^{k} p_{1,u}$ equals zero. The second part of equation (4.4) computes the sum of the transition probabilities between the state $\omega_{1,1}$ and all states in the other $(n-1)$ dimensions. The part $\sum_{u=k+1}^{kn} p_{1,u}$ is equal to:

$$\sum_{e=2}^{n} \sum_{u=(e-1)k+1}^{ek} \frac{\|\omega_{u,e} \cap \omega_{1,1}\|}{\|\omega_{1,1}\|(n-1)} = \sum_{e=2}^{n} \frac{\sum_{u=(e-1)k+1}^{ek} \|\omega_{u,e} \cap \omega_{1,1}\|}{\|\omega_{1,1}\|(n-1)} = \sum_{e=2}^{n} \frac{\|\omega_{1,1}\|}{\|\omega_{1,1}\|(n-1)} = 1 \tag{4.5}$$

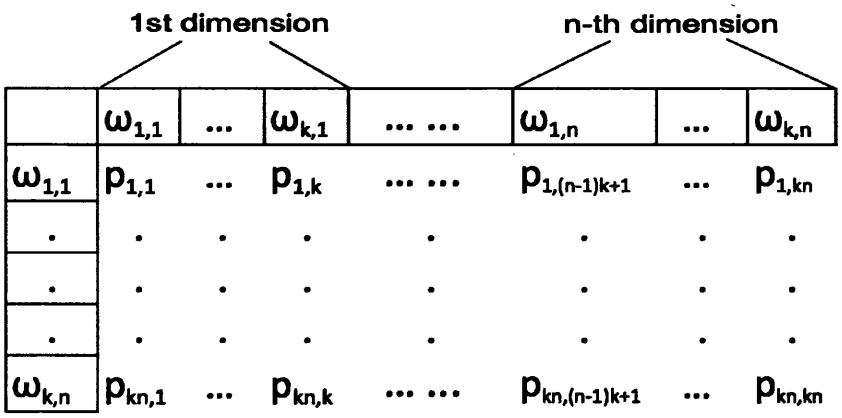


Figure 4.3: This image shows a $kn \times kn$ transition probability matrix where k is the number of bins or states in each dimension and n is the number of dimensions.

Then the results in equations (4.4) equal unity. In this chapter, the probability value is mainly used for ranking. According to equation (4.2), we learn that the degree of probability value for each data sample indicates the number of patterns which are similar to it. If two data samples are passing through a similar set of states or bins in a Markov Chain, then we say these two data samples have a similar profile in multidimensional space.

4.2.2 Data Preprocessing

The transition probability of the Markov Chain is based on the line-segment frequency between the bins in neighboring axes, namely $\|\omega_{i,j} \cap \omega_{u,v}\|$. In order to enable fast data exploration, we pre-compute a transition probability matrix and store it in an external file. For n axes and k uniform intervals, the total number of bins to be computed and stored is $\frac{n(n-1)k^2}{2}$. Whenever a new Markov Chain is determined by the user, n reads suffice to compute a joint probability value. In addition to the transition probability, we also pre-compute a bin map which stores the data frequency in each dimension, namely $\|\omega_{i,j}\|$, which requires kn bins in total. Throughout this chapter, we set the number of bins to be 64, utilizing this we are able to obtain clear results for different data sets. When rendering a high-dimensional data set, the pre-computation for a transition probability matrix might require a longer time. A detailed bin map (256*256 bin counts) takes up 256 KB of memory when four bytes are used per bin count. A dataset with 50 dimensions requires approximately 300 MB to store the bin maps for all possible axis-axis combinations. However, once a matrix is ready then computing a probability distribution for all data samples will be fast, which only requires nm additions where m is the size of the data set. The number of bins is determined by the user depending on how much abstraction they would like. Too large a bin width might lead to cluster with a large diameter but suffers from large variance. Too small a bin width might cause over-fitting so that most of the probability values are near zero. Our solution is to provide interaction support for the user to change and

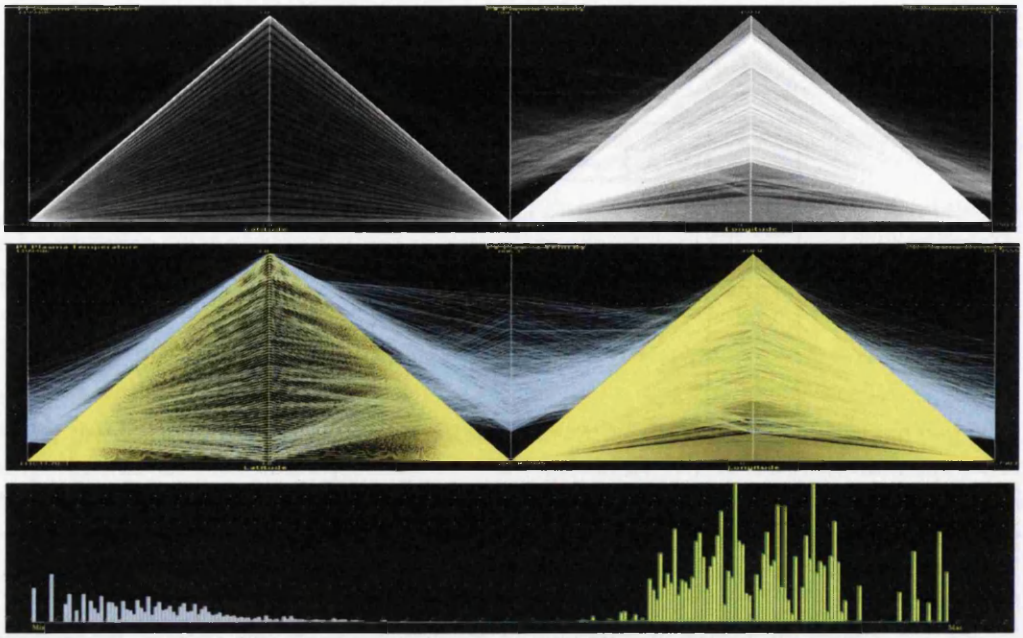


Figure 4.4: This image shows a NASA Mission data set obtained from the XMDV web page [XMD11]. This data set has 7 dimensions and 8784 records. The first row shows the line-based histogram. The second row shows the composite brushing using our method. The third row shows the probability histogram.

experiment with various bin widths for different data sets. Theoretically there is no limit for the number of bins selected because the binning is performed in data space. However the shape of the data trend in the output visualization is constrained by the number of pixels on the screen. Therefore we suggest for the number of bins that it is best not to exceed the number of available pixels in the screen space. Also too great a number of bins results in an enormous transition probability matrix.

4.3 Visualization and Analysis

Once we have obtained a list of probability values for all data items, this can be represented either by a scatterplot or a histogram. In our paper, a scatterplot can be used when rendering a small number of data items, whereas a histogram is adopted when rendering a large data set.

4.3.1 Scatterplot Representation

In the first example, we would like to consider a synthetic dataset about the geometric features of pollen grains consisting of 3848 observations with 5 variables. This is the 1986 ASA Data Exposition data set from David Coleman of RCA Labs [WL97]. From the probability scatterplot shown in the third row of Figure 4.5, we can see that most of the data items have relatively

low probability values as depicted in blue. However there are 99 data items having a much higher probability and are isolated from the original 3848 points as depicted in yellow. If we render these data trends and noise separately on parallel coordinates in different colors, we obtain the visualization on the second row in Figure 4.5. As we re-scale the selected data trends drawn in yellow, we are able to catch six clusters, as shown in the fourth row of Figure 4.5. Although the extracted 99 data points are only approximately 2.7% of the data set, we are able to successfully isolate these points from the noise. In the previous method [WL97], the time cost to prune the noisy data is approximately within three minutes. However, in our method the identification occurs instantly once we have computed the joint probability distribution. The first row of Figure 4.5 is a visualization using the line-based histogram [NH06]. The patterns in the middle of the axes are not very sailent and clear. In addition, we are unable to extract and separate these data points out of the noise by this method.

4.3.2 Histogram Representation

When rendering a larger data set, its probability distribution is normally represented by a histogram. In this example, we are using a NASA Mission data set obtained from the XMDV web page [XMD11]. This data set contains 7 dimensions and 8784 records. From the probability distribution shown at the bottom of Figure 4.4, we can easily discern a gap between the high and low probability range. If we render them separately on parallel coordinates in different colors, we are able to obtain a visualization shown in the second row of Figure 4.4. The principal data trend depicted in yellow is perfectly isolated with the remaining data samples depicted in blue. The top row of Figure 4.4 is rendered using the line-based histogram. Although it can also reveal two triangular shapes in two subspaces, there is discontinuity along the middle axis. According to the previous work, this discontinuity can be explained as uncertainty [FKLT10]. However, it is local uncertainty only in two-dimensional subspace, not in the n-dimensional space. The blue patterns displayed in the second row of Figure 4.4 show uncertainty in multi-dimensional space relative to the principal data trend depicted in yellow. The discrete histogram view is fast to compute and can represent a general data distribution. If the user favors a smoother data distribution representation, a Gaussian Mixture Model can be applied but with a high computational overhead.

4.3.3 Data Rescaling

Once a subset of the data set has been brushed, we provide a user option to re-scale the selected data items. The re-scaled data might uncover some of the interesting patterns. In Section 4.3.1, when we further re-scale the brushed data items, we are able to catch six clear clusters. The other example we demonstrate is the stock market data. The top row of Figure 4.6 shows the daily total volume, open price, close price, the highest and lowest value of transaction in the NASDAQ stock market from 1970 to 2010 [Inf11]. If we brush the high probability range and re-scale the selected data items, we can observe the inherent data trend structure as shown in the second row of Figure 4.6. Over 91.2% of the data set is brushed. From the color coded polylines, we can observe a strong correlation in the first four attributes.

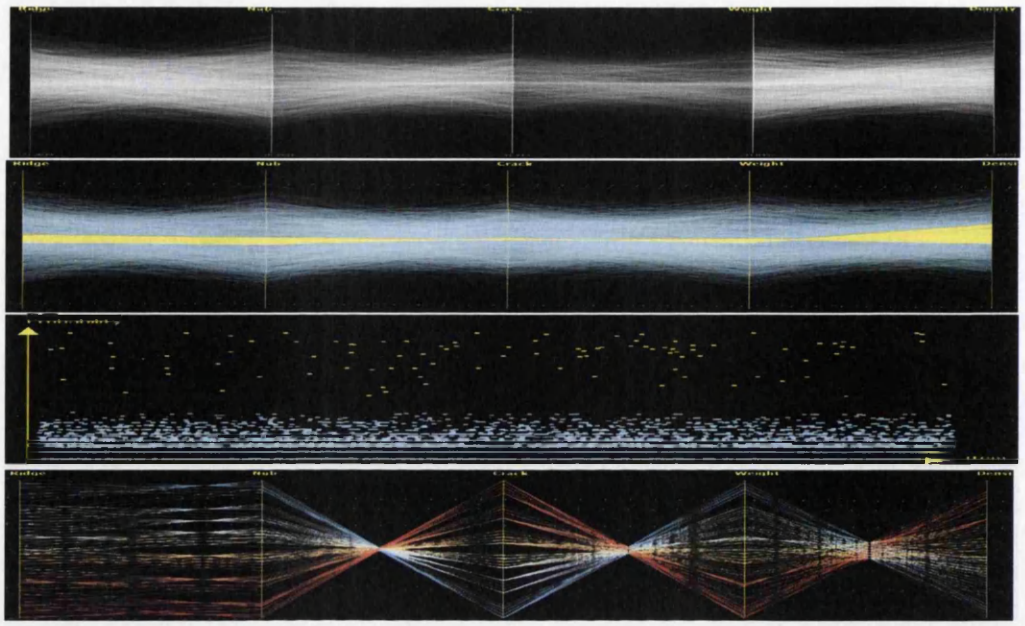


Figure 4.5: This figure shows the pollen data set. The first row shows the line based histogram. The second row shows the composite brushing, with the yellow polylines showing the trend and the blue polylines the noise. The third row shows the scatterplot of probability distribution. The fourth row shows the re-scaled visualization.

4.3.4 Probability-Based Brushing

The examples used in Section 4.3.1 and 4.3.2 illustrate the patterns which are easy to discern and separate. However for most of the data sets, their probability distributions do not permit a clear separation of the data trends. For a large data set, our goal is to select a subset of the data which can mostly represent multi-dimensional features and characteristics. In this section, we introduce some of the brushing techniques to handle this class of data sets. The essence of our technique is that a data sample in a multidimensional principal trend will have a high joint probability value, whereas a data item regarded as an outlier often has a very low probability value. The question arises as to how to classify the high, medium and low probability ranges. Once we are able to label these classes for each data sample, then a composite brushing can be performed. Brushing data samples classified as high probability will capture the principal trends, whereas data samples classified as low probability show a minor trend or outliers. As for the rest of the data samples in a medium probability range, we provide two user options to either filter these data samples out or render them as a context view.

4.3.5 K-means Classification

In our work, we provide a classification as guidance for the user to brush for data patterns. The user can determine whether to apply our classification for automatic brushing, or to directly

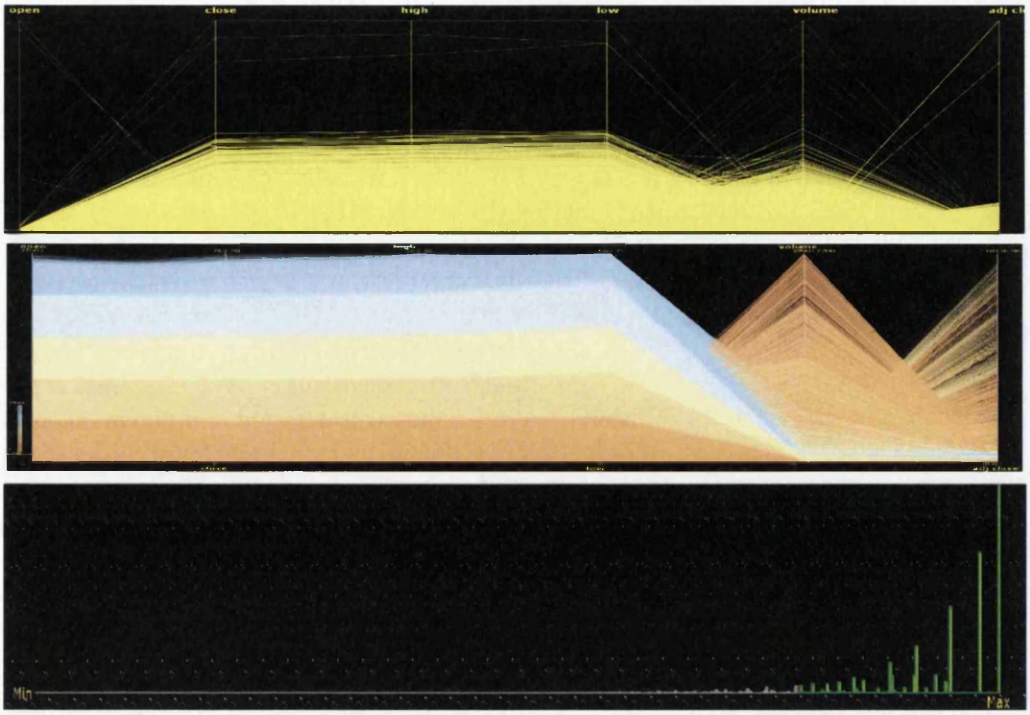


Figure 4.6: This data represents the daily volume of transactions, the opening price, the closing price, the highest and lowest volume of transactions in the NASDAQ stock market from 1970 to 2010 [Inf11]. There are 838,582 data samples. We see the original parallel coordinates (top row); the brushed and rescaled polylines (bottom row).

interact with the probability histogram. Initially, we compute the 10^{th} percentile, median and 90^{th} percentile values of the probability lists of all data samples. From this information, the user is able to brush these arbitrary partitions for different trends. If they favor automatic brushing, then we use these values as the initial means for K-means clustering. A threshold is set by the user to determine how many iterations are desired for a mean value update. We set the number of iterations to 15 as the default for fast computation. Because the K-means clustering operates on our one-dimensional probability list, it is reasonably fast and efficient to compute the euclidean distance between data items. The number of clusters namely k, is set to three as default referring to "high", "medium" and "low" probability range. In addition, we allow the user to choose an arbitrary number of clusters for their desired classification. One point we emphasize is that the classification based on this probability distribution is used for data extraction and brushing, rather than data clustering. A high probability range indicates global principal trends and a low probability range presents minor trends or outliers.

4.3.6 Composite Brushing

In this example, we consider a real biodiversity informatics data set. This data set is from a clustering-based niche envelope model that William Hargrove and Forrest Hoffman studied for Lobolly pine across the contiguous United States [HH00]. From this data they aim to classify the Lobolly pine based on twenty-five factors, including elevation (ELEV), maximum, mean and minimum annual temperature (MAXANN, MEANANN, MINANN), monthly precipitation (PCPJAN to PCPDEC), several soil parameters, number of frost-free days (FFREE) and solar output and input. Each data element represents a data map which was developed for the continental United States at a resolution of 1 km^2. This data set has 49324 samples. It can be downloaded from the XMDV website [XMD11]. Shown in the bottom of Figure 4.9 is a histogram representation for the probability distribution. The histogram is partitioned into three parts using K-means clustering. Each class is depicted in different colours. If we brush the high probability cluster (depicted in yellow) and low probability cluster (depicted in blue) respectively, we are able to obtain a visualization shown in the fourth row of Figure 4.9. The pattern in yellow illustrates a principal data trend. There are 45.5% of the data samples brushed which reveals a coherent pattern propagating through twenty five dimensions. In these areas, the overall annual temperature is moderate as shown in the first three axes. From axis ELEV we can see these areas are low in elevation. The precipitation has larger variance from January to May than from June to October. In addition, the precipitation drops from May and remains relatively low in the next few months. The depth to water table (WDEPTH) remains high.

4.4 Comparison

In this section, we compare our results with other popular large data visualization techniques, such as alpha blending [Weg90a, WL97], hierarchical clustering [FWR99] and line-based histograms [NH06, RK08]. The first row in Figure 4.9 is a visualization rendered using alpha blending. The density of the plots is represented with transparency. Under a low alpha value, the sparse parts of the dataset fade away while the more dense areas are emphasized. This works well with small datasets, however, with large datasets the range of data is much greater and consequently it is more difficult to fully represent the fidelity of complex datasets. It is difficult to obtain a clear understanding of patterns and clusters, as it becomes cluttered in some areas between axis AWC150 and ELEV. The yellow patterns in our method as shown in the fourth row of Figure 4.9 provides a much clearer data trend. In addition, outliers may get lost using alpha blending, such as patterns on the bottom between axis PCPJAN and PCPDEC. Our method is able to preserve such outliers as depicted in red patterns. If we combine the yellow and red patterns in our method, we are able to approximately reconstruct a complete view of data features with an emphasized view of principal data trends.

The second row of Figure 4.9 shows a hierarchical parallel coordinates rendered by XMDV [FWR99]. In this approach, a Birch's hierarchical clustering algorithm is adopted which builds a tree of nested clusters based on proximity information. Proximity-based coloring is introduced to demonstrate clusters, and transparency to show the mean and the extent of each cluster. Then multi-resolution views of the data can be rendered. Compared with the alpha blending, it is able to offer a clearer data distribution and preserve the low frequency data

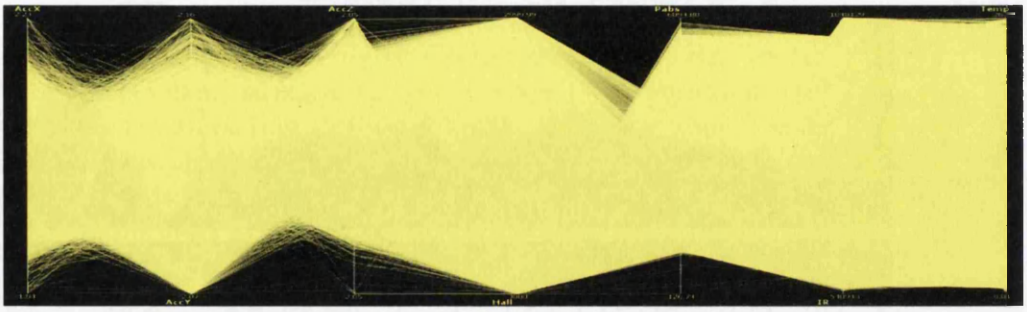


Figure 4.7: This figure shows the original parallel coordinates on animal tracking data set. It suffers from heavy overplotting.

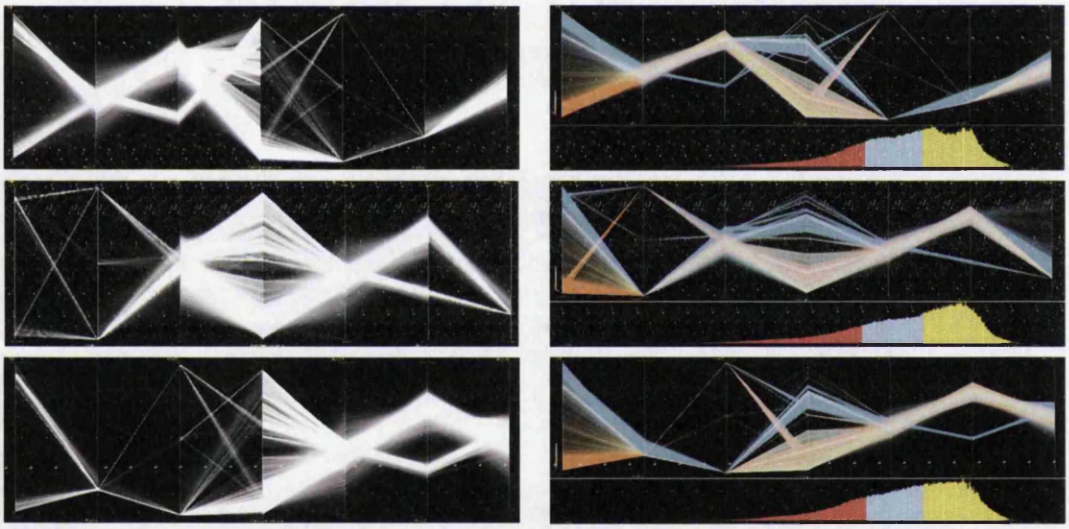


Figure 4.8: This figure shows the visualizations of three different orderings of our animal tracking data set. The data patterns on the left column are rendered using a line-based histogram, while the patterns on the right column is the brushed data samples with high probability using our method. We note that a color scale is mapped to the position of polylines according to the first vertical axis in the parallel coordinates view to depict the coherent patterns in higher dimensions.

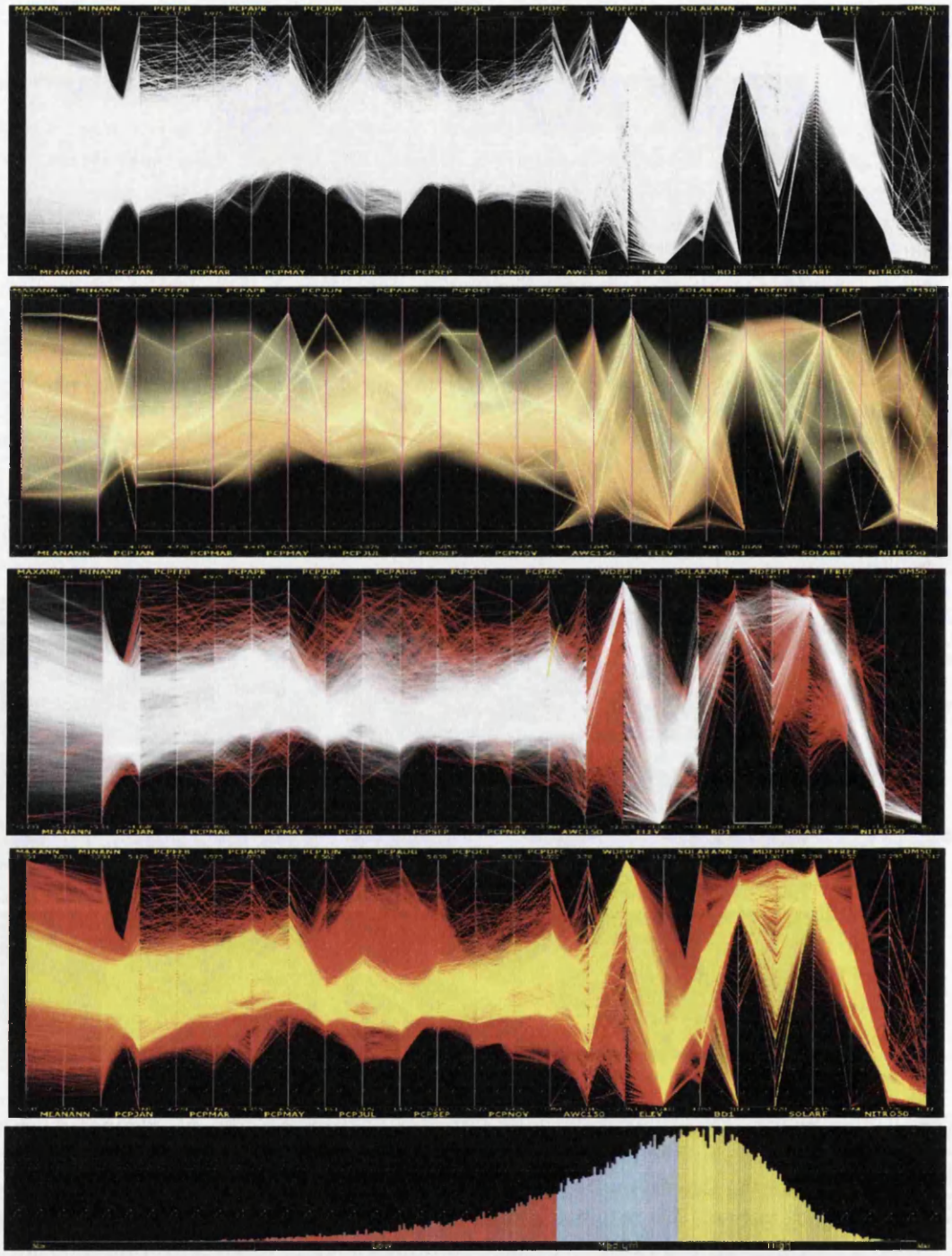


Figure 4.9: This figure shows a proprietary, biodiversity data set from XMDV [XMD11]. This data set has 25 dimensions and 49324 samples. The first row is visualization using alpha blending. The second row shows the visualization rendered in hierarchical clustering and proximity-based representation. The third row shows the outlier-preserving line-based histogram [NH06]. The fourth row shows our composite brushing with yellow patterns representing a high probability range and red patterns a low probability range. The fifth row shows our probability histogram.

samples. However, the densities of different clusters might be difficult to distinguish from XMDV. This is because most of polylines are rendered in a small portion of screen space, the differences in the proximity-based transparencies for various levels of clusters are not easy to discern. Using our method, we are able to immediately catch a clear view of central data trends as depicted in yellow and minor data trends depicted in red. In addition, because each polyline rendered is a mean value of a cluster in XMDV, they may deviate from the original positions of the polylines and cause problems with interpretation. For example, outliers on the bottom between the axis PCPJAN and PCPDEC are supposed to be at minimal value of each axis and patterns on the lower part between axes ELEV and SOLARANN are different from patterns in alpha blending and our method.

The third row of Figure 4.9 shows a visualization rendered using line-based histograms [NH06, RK08]. It is built upon a two-dimensional bin map storing the frequency of line segments in every neighboring dimensions, which is similar to our method. Eventually, every bin is rendered as a parallelogram connecting a pair of intervals at adjacent axes with its vertexes placed at the respective positions of the minimum and maximum bin borders with its frequency represented by transparency. Then the high frequency histograms are emphasized whereas the low frequency ones fade away. Although the local data trends can be discovered from this method, the global trends in higher dimensions are missing. This causes discontinuity across high dimensional space, as highlighted in yellow. Our method is advantageous by overcoming such discontinuity by offering a coherent global data trend in multi-dimensional space. The previous method [NH06] also proposes an outlier detection method. For any low frequency bin, a 3 by 3 isolation filter is used to check the occupancy values of the 8 bins that are adjacent to the central bin. If the number of empty neighbouring bins is above a certain threshold (say 2 for the corners, 4 for the borders and 6 or 7 for the rest), the central bin is declared an outlier which is shown as the red patterns in the third row of Figure 4.9. As we can see, the outliers obtained are limited to a two-dimensional subspace which lose the continuity in higher dimensions. However, our method is able to present a multi-dimensional outliers by brushing a low probability range.

4.5 Markov Chains Manipulation

A joint probability value for each data sample is obtained by summing up all of the transition probabilities between pairwise dimensions. Every time the dimension order is changed, the probability distribution has to be re-computed. This is also true of the previous line-based histograms [NH06, RK08].

4.5.1 Dimension Re-ordering

As our chapter title implies, the focus of this chapter is on visual pattern analysis. Since reordering the parallel coordinates often leads to different visual patterns, therefore our probability distribution is optimized for these orderings in order to achieve the best visual effect. In section 4.2, we pre-compute a transition probability matrix and store it in an external file. Based on this matrix, computing a new probability value for each re-ordered data tuple is very fast, which requires n additions where n is the number of dimensions. Because our initiative is

to extract the major patterns in visual space, therefore the change of individual element probability value in data space is of no interest in this chapter. As long as the probability distribution is optimized, we can always display data samples with highest probability values to form the principal visual patterns in screen space to overcome visual clutter. This is the same as the previous output-oriented line-based histogram [NH06]. The difference to the previous approach is that we consider n-dimensional coherent visual patterns rather than two dimensional discontinuous visual patterns.

In section 4.2.2, we pre-compute a transition probability matrix and store it in an external file. Based on this matrix, computing a new probability value for each re-ordered data tuple is very fast, which requires n additions where n is the number of dimensions. After the new probability values are generated, K-means clustering is used to classify high and low probability ranges for automatic brushing. No matter how the dimension ordering changes, we can always obtain a group of most representative data samples with highest probability values. Therefore the principal data trends observed in one ordering will not disappear in other dimension orderings. In this example, we consider a real world marine biology data set [GJL+09]. Biologists at Swansea university have collected a large amount of data relating to animal movement by attaching sensors to individual subjects. The data here is re-sampled once a second over five days. In this example, we select 7 important data attributes with 536,548 records. This data set can be plotted using traditional parallel coordinates, but suffers from heavy overplotting, as shown in the top image of Figure 4.7. Shown in Figure 4.8 are the different orderings of our marine biology data set. Visualizations on the left column are rendered using the traditional line-based histogram and on the right column are the polylines brushed by the high-probability data items using our method. In our method, each polyline is assigned a different color along a user-defined axis, in our case the first axis (IR) is chosen. As we can see, the line-based histogram is able to illustrate local data trends in a two dimensional subspace, whereas our method manages to capture global data trends in higher dimensions. Then we are able to observe relationships and dependencies between any dimensions rather than in neighbouring dimensions in previous method. From the probability histograms, we can see that different dimension orderings lead to different probability distributions. However, three probability ranges depicted in different colours in a histogram view can be automatically generated by the K-means clustering each time the dimension ordering is changed. As we brush data samples with a high-probability range (depicted in yellow in the histogram view), we can always obtain principal data trends shown on the right of Figure 4.8.

4.5.2 Subspace Pattern Discovery

In the previous sections, we have incorporated all data dimensions to form a Markov Chain for each data tuple. However sometimes the user needs to see the patterns in a subset of the dimensions. In Cagatay Turkay et al.'s work [TFH11], they point out that it is important to make the user understand the structure of the dimension space as well as the data distributions with respect to the dimensions. In order to help the user manipulate different structures in the dimension space, we provide a user option to change the number of dimensions to generate a new sequence. Different from the previous line-based histogram [NH06, RK08], our method is able to capture the patterns in any arbitrary subspace. Based on the pre-computed transition

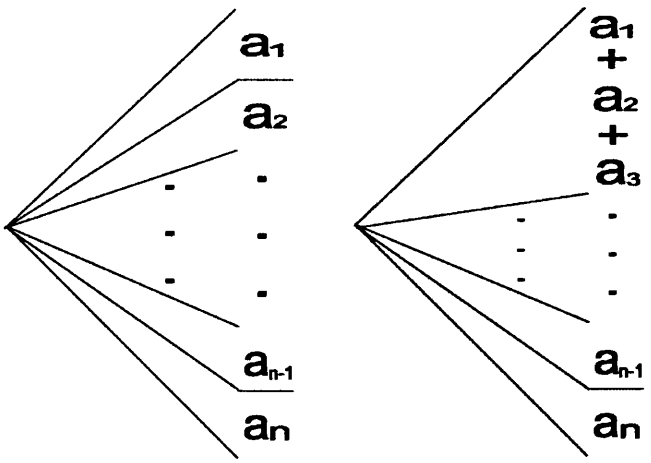


Figure 4.10: Shown on the left, is our angular splitting scheme. For each histogram bin, we split an angular range for a histogram bin along an adjacent axis into into n groups, where $a_1, a_2, ..., a_n$ are the data frequencies in each angular partition. In order to reduce the number of histogram bars rendered on the screen to avoid visual clutter, we further propose an adaptive angular representation using K-nearest-neighbor optimization as shown on the right. We assume that the frequencies in the first three angular partitions are above a threshold, therefore these three contiguous partitions are merged. A new average angle is computed.

probability matrix, the probability in any sequence of Markov Chains can be quickly computed. During the rendering process, we still render an n-dimensional polyline each time regardless of how many dimensions are considered in the chain. This allows the user to keep track of the relations between the chains and the other unselected dimensions.

4.6 Scalability to Large Data

In this section we discuss our solution when a brushed portion of a data set is large. Fua et al. define large data sets as containing 10^6 - 10^9 data elements or more [FWR99]. When rendering a large data set, such as the animal tracking data discussed in Section 4.5, the size of brushed data samples might become large which brings two problems. First, too many lines rendered on a limited pixel space will lead to a clutter such that the data patterns cannot be distinguished. Second, even the brushed patterns can be discerned, the inherent data density of line plots in these patterns might not be easily uncovered.

4.6.1 Alpha Blending

One way to address this problem is to apply alpha blending. As shown on the right column of Figure 4.8, the brushed data trends are rendered using a low alpha value. In these visualizations, approximately 53.5% of the animal tracking data set containing 28,726 data samples is rendered. These brushed data samples illustrate the global trends in higher dimensions. Each

coherent pattern can be clearly distinguishied in different colors by the utilization of alpha blending. However, because the resolution of the alpha channel is soon exhausted, the density of the revealed patterns cannot be easily discerned. In this section, we propose an extended angular histogram to address this problem and scale our method to large data sets.

4.6.2 Extended Angular Histogram

The angular histogram was proposed and presented in the last chapter [GPL+11]. In this approach, each polyline-axis intersection is considered as a vector. The angular histogram can be used to aggregate both the magnitude and direction of these vectors. The basic idea is to calculate for each histogram the mean angle of the vectors and rotate histogram bar by this angle. Shown on the top of Figure 4.11 is a traditional angular histogram applied on our brushed data samples of the animal tracking data. When the histogram bars are rotated by a given angle, it's more difficult to discern and compare their relative lengths. Therefore we apply a colour map to the histogram bars to represent the data density. Although the average angle of each bin is a representation of the central tendency, it can be sensitive to extreme values (e.g. outliers) and the standard deviation might become significant. In order to accurately display the profile of the data trend, we further proposed a divided angular histogram in last chapter which splits the histogram bin into two separate groups, one containing vectors with an upward slope and the other with a downward slope. We have two concerns with this approach: First, it requires massive user interactions to tune the best set of parameters to avoid the visual clutter caused by excessive histogram bars. Second, only considering two groups of angular ranges may cause the lack of precision when representing a multi-modal angular distribution. In this section, we propose a more precise angular approximation for angular histograms. Our extended angular histogram can be further applied to our brushed data samples for data trend density illustration.

4.6.2.1 Angular Subrange Partition

In order to provide a precise angular approximation, we split an angular range for a histogram bin along an adjacent axis into thirty-two subranges as opposed to two in the previous method [GPL+11]. The user is also able to specify their desired number of angular partitions. Each partition represents an angular subrange, slopes of line segments within this subrange will be aggregated. The left of Figure 4.10 demonstrates how our angular partitioning works. For each direction in a bin, we summarize the frequency of the line segments in this direction. This frequency determines the length of a histogram bar pointing to this angle. As a result, every bin along one axis requires thirty-two histogram bars with each one representing the direction and frequency of one angular subrange. This process generates an enormous number of histogram bars to be rendered which might cause serious overplotting on the screen. In order to address this problem, we further propose a K-nearest-neighbor approach to optimize the number of histogram bars displayed. This is discussed in the next section.

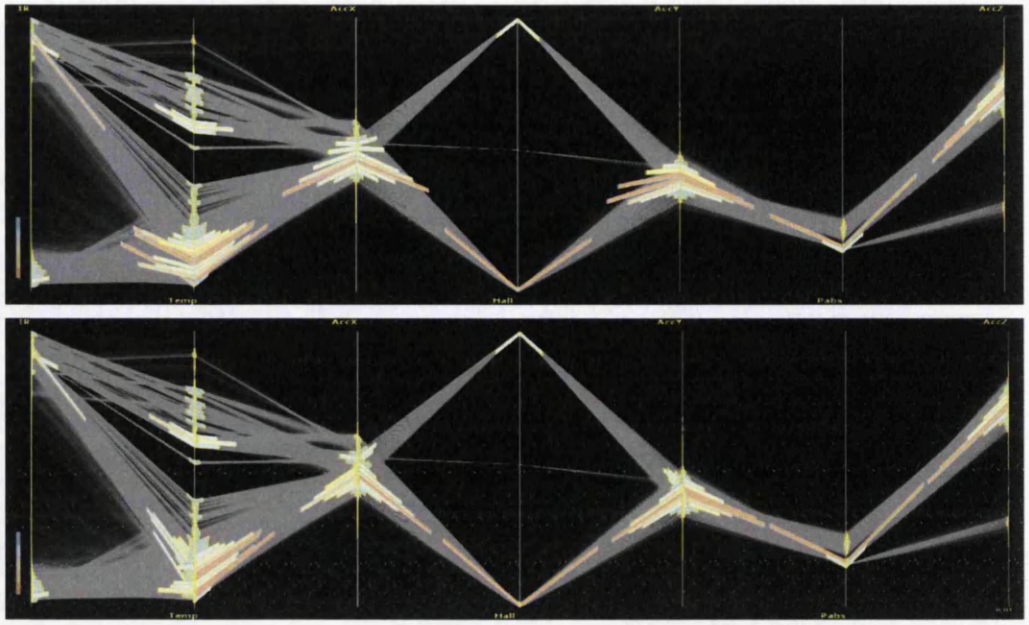


Figure 4.11: This figure shows the traditional angular histogram with each bar rotated by an average angle (top); our angular histogram with adaptive angular representation (bottom).

4.6.2.2 K-Nearest-Neighbour Optimization

This approach is based on the fact that if the frequency of the lines in the current angular partition and in its k nearest neighbouring partitions are all high, these partitions can be merged into one and represented by a single histogram bar. The algorithm can be explained as follows: For each histogram bin, we iterate through its angular partitions. For each partition, we examine if the line frequency in this partition is above a threshold ξ. If so, we continue to test its neighbouring partitions. If the frequency of its neighbouring partition is also above the same threshold ξ, then two partitions will be merged. This process will continue to the next k contiguous partitions until the termination criteria is met. A new average angle is computed for these merged angular partitions. We define a termination criteria as the frequency of the examined angular partition is below a user defined threshold ξ. When this occurs, a histogram bar pointing to current merged angular direction is rendered. We set the initial K value to eight. After this optimization, the number of angular partitions for each bin is reduced. Shown in the bottom of Figure 4.11 is our improved angular histogram. Compared with the previous method shown on the top, we are able to observe that our method offers a more precise angular representation. The angular histograms of the first ordering of Figure 4.8 are shown in Figure 4.12. The underlying polylines are the brushed data trends rendered using a low alpha value. As we can see, the angular histograms can not only depict the density distribution of the data samples, it also reveals where the data trends flow.

We also provide a user option to change the width of each angular histogram either globally or locally. Global bin selection applies the bin width to all histograms across all axes. Whereas

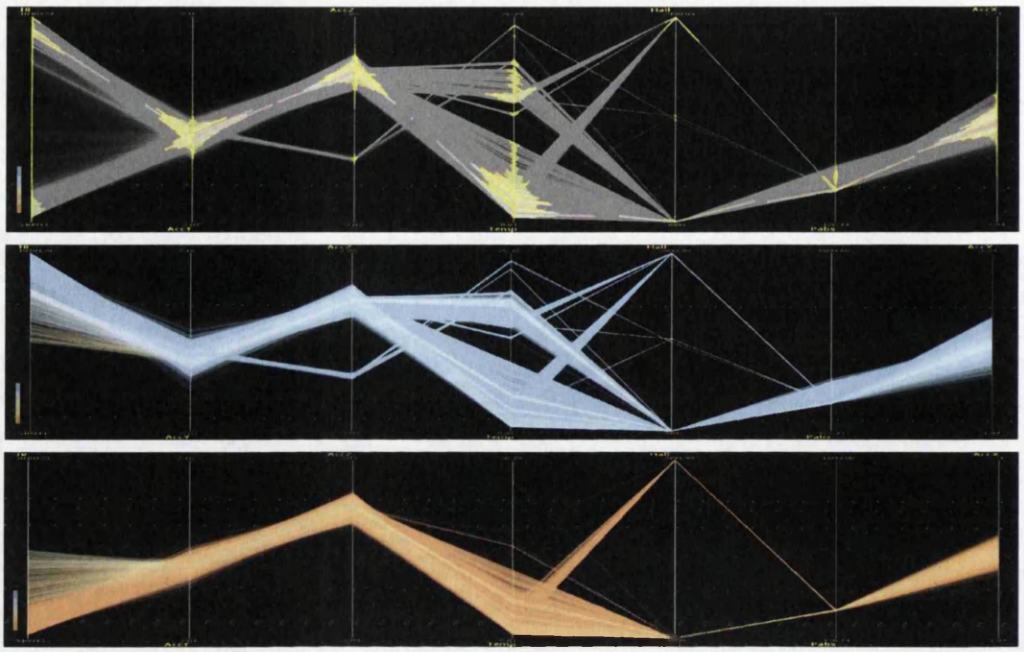


Figure 4.12: The top row shows the angular histograms imposed on the brushed data trends in our marine biology data set. The middle and bottom row is the divided patterns by the first axis (IR) from the brushed polylines. A complete color-coded view is shown on the top of right column in Figure 4.8.

local bin selection allows the user to adaptively select the bin size in different areas. For example, the areas with high data density might require a smaller bin width and thus more bins to depict finer detail. In order to quickly compute the multi-resolution global bin width, we always set the initial number of bins to be a power of two. When increasing the bin width, we merge the neighboring bins together and there is no need to re-compute the bin distribution.

4.7 Use Cases

In this section, we provide an in-depth analysis of the data trend discussed in Section 4.5 on our marine biology data set. As mentioned in the previous section, there are seven measurements in this marine biology data set which can be described as:

- Tri-axial accelerometer data in x, y and z: Accelerometers measure acceleration forces (g) by reacting to changes in the earths gravitational field.
- Hall: By positioning a Hall sensor on the upper mandible of the turtle and a neodymium boron magnet on the lower mandible, it is able to record when the turtle opens its mouth.
- Pressure: A measure of the force per unit area perpendicular to the sensor.

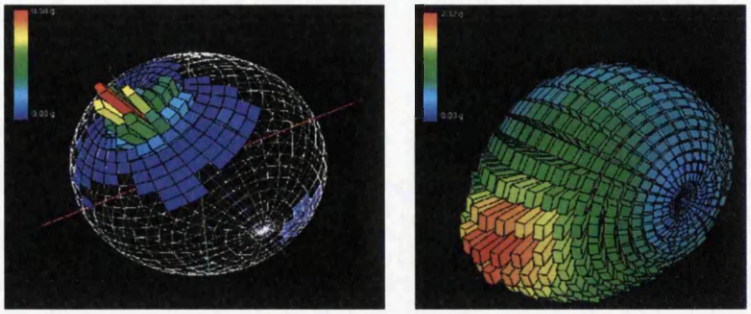


Figure 4.13: Spherical scatter plots are used to show the geometric distribution of data, and spherical histograms show common animal movements [GJL+09]. This figure shows spherical histograms of an accelerometer in X,Y,Z directions. Shown on the left is our brushed data samples and on the right is the total data samples. Color is mapped to histogram frequency.

- IR: A measure of electromagnetic radiation (mV). As water flows over the paddle it is bent backwards indicating the relative speed of an animal in water [LPL+08].
- Temperature: Measures the external environmental temperature.

As shown in the first row of the right hand column in Figure 4.8, two major clusters along the first axis (IR) are revealed in different colors. If we decompose these two clusters by the first axis and render them separately, we are able to obtain the data patterns shown in Figure 4.12. The second row of Figure 4.12 shows the data trends with high IR value and the bottom row shows the trends with low IR value. The IR value indicates a relative speed of the animals moving against water, high IR often suggests a low speed and low IR indicates a high speed [LPL+08]. From the angular histogram shown in the top of Figure 4.12, we are able to see that the size of patterns with high IR are much larger than the patterns with low IR. This indicates that most of the time the animal is moving in a low speed in our brushed time steps. When the relative speed is low, as shown in the second row of Figure 4.12, there are two patterns can be observed with one leading to high temperature and the other to low temperature. Because the deeper the animal is swimming in the water, the lower the temperature. Combined with the pressure value (Pabs), we can infer two actions. One is that the animal is diving into the water and the other is ascent up to the surface. Moreover, from the mouth sensor (Hall), we can observe two patterns, one is that the animal closes its mouth when it is moving and the other is it opens mouth when preying. The data trends shown in the bottom of Figure 4.12 is a cluster with low IR value, which means that the animal is moving at a high speed. This pattern mostly leads to low temperature, from which we can infer that the animal is diving or swimming quickly under the water. We can also map our accelerometer in x, y, z to a spherical histogram [GJL+09], which is intuitive for the user to observe common directions of the animal movements as shown in Figure 4.13. From this visualization, we are able to see that an animal is moving toward a certain range of directions in most of the brushed time steps. Compared with the line-based histogram shown in the first row of left column in Figure 4.8, our method preserves salient global features of the data, uncovers different clusters and avoids

pattern discontinuity across n-dimensions. By exploring the angular histograms, we are able to observe the data density in the brushed data patterns, especially in a large data set. We note that a same color scale from ColorBrewer [BH06] is mapped to the position along the first axis for polylines and to the frequency in each bin in angular histogram view respectively.

Chapter 5

Visualizing Translation Variation of *Othello* : A Survey of Text Visualization and Analysis Tools

Contents

5.1 Introduction

This chapter is based on a publication from Geng et al. [GSC+12]. William Shakespeare's plays have been translated into every major living language. In some languages, his plays have been re-translated many times. These translations and re-translations have been produced for about 250 years, in varying formats: some as books, including reading editions and study editions; some as scripts for performances (theatre, film, radio and television scripts). Multiple heritage text translations have remained, until now, an untapped resource for Digital Humanities. Divergence of multiple kinds caused by multiple factors is normal among multiple translations, due to differing translation purposes, genetic relations (translators 'borrowing' from one another), context-specific ideological and cultural influences, inter-translator rivalry, and translator competence and style. Studying variations in re-translations of world cultural heritage texts is of cross-cultural interest for humanities researchers. This does not just apply to Shakespeare. Variations among re-translations reveal histories of language and culture, intercultural dynamics, and changing interpretations of every translated work.

The interpretation of Shakespeare's work in translation is always influenced by the translator's own culture, customs and conventions. Therefore, each translation is a product of changing culture as well as an expression of each translator's individual thought within that culture. Also, each translation is a reply to received ideas about what Shakespeare's work means. Semantic and textual variations between translations in the corpus carry relational cultural significance. Normally, researchers from Arts and Humanities read and compare cultural text in its raw form and this makes the analysis of the multiple translations difficult. In addition, interesting patterns are often associated with text metadata, such as historical period, place, text genre or translator profession.

Digital Humanities researchers working on a project called 'Translation Arrays: Version Variation Visualization', have collected an experimental corpus of fifty-five different German re-translations of Shakespeare's play *Othello* (1604). The translations date from between 1766 and 2010. Most texts were acquired in non-digital formats. A representative sample of 32 of the re-translations has been digitized. The 32 texts of one scene of the play have been cleaned, formatting normalized, all texts segmented, speech by speech, and all segments semi-automatically aligned with a so-called 'base text' (Shakespeare in English), to create a parallel corpus. The selected scene is Act 1, Scene 3: in Shakespeare's original text, this scene is c. 10% of the play's length; it has c.3,000 words from the play's total of c.28,000 words; and the scene has 88 speeches. This parallel corpus can be accessed at the Translation Arrays project website: *www.delightedbeauty.org/vvv*. Based on this corpus, the team want to explore variations between different translations at the segment level, in order to uncover patterns relating to different types of translation, historical periods, genetic relations, and patterns relating to different sub-sets of segments. Sub-sets include speeches by certain characters (with the hypothesis that translators interpret characters in the play in distinctive ways, and therefore translate their speeches in different ways), and segments with certain linguistic and poetic features, such as metaphors, puns, rhyme, interpretative challenges, and so on. The team's general long-term aim is to develop analytic tools which will work for any corpus of re-translations. In this chapter, the domain experts have selected a subset of their collected translations which are of great interest and they would like to analyze and explore the varations between them. The detailed information of these selected documents are discussed in Section 7.3.

The goal of this chapter is to survey and compare free text visualization tools and to visualize the various translations of Shakespeare's work, *Othello*. The initial task is to identify and extract the non-semantic features from the texts within a document corpus. The non-semantic features refer to the number of words, tokens and patterns in the concordance. Text pre-processing facilitates the construction of text concordance, term relations, document relevance and other properties of interest. Based on the extracted text attributes, various visualizations can be applied. In this document, we present the results of our survey on the state-of-the-art techniques and free, off-the-shelf tools for text analysis and visualization. We conduct some experiments with a selection of tools using Shakespeare's *Othello* as an example. We investigate if any of the freely available tools can provide clues to the variation in German translations of *Othello*.

The rest of the chapter is organized as follows: Section 5.2 briefly introduces the related projects of the visualization applications for specific text corpus. Section 5.5 investigates different research prototypes and freely, available visualization softwares. Section 5.6 compares

our investigated state-of-the-art visualizations based on the visual mapping and interaction design. Section 5.7 introduces our proposed, customized visualization techniques.

5.2 Related Work

Visualizations refers to an emergent body of computational work which uses digital graphics to help people understand and explore complex data-sets. The best of this work combines artistic design and intuitive ease of interactive use. It allows us to understand important subject-matter in new ways, by literally seeing it in new ways. Excellent examples which are free to view online include:

- Work by Hans Rosling which presents information on global health economics - `www.gapminder.org(2008ff.)`
- Work by Ben Fry which presents Darwins successive redactions of On the Origin of Species - `http://benfry.com/traces/` (2009) - and last but not least,
- Work by Stephan Thiel which presents all the plays of Shakespeare, using the "deeply tagged" WordHoard digital texts, filtered through analytic algorithms www.understanding shakespeare.com (2010)

These visualizations are specifically made for one application and the users are unable to upload their own data sets for further analysis. In this chapter, we mainly concentrate on how emerging state-of-the-art visualization research prototypes and freely available software can benefit our study on German translations of *Othello*.

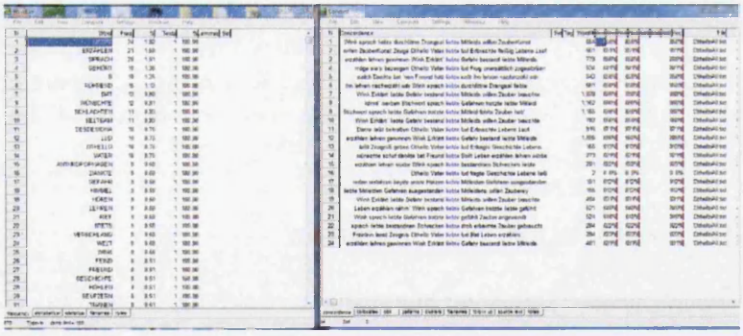


Figure 5.1: This figure shows the interface of WordSmith developed by [Wor96].

5.3 Text Preprocessing

Using the text preprocessing tools introduced in this section, we can collect a wide range of text attributes, such as word relationships, word frequency and sentence segmentation. Domain experts are particularly interested in the variations in different segments or paragraphs of several documents.

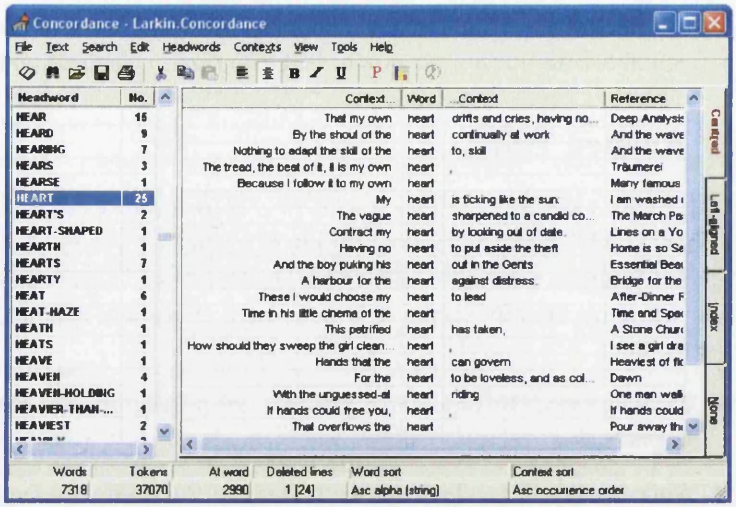


Figure 5.2: This figure shows the interface of Concordance developed by [Wat09].

The software WordSmith, developed by [Wor96], is able to generate various text attributes, such as word frequency, parts of speech and other statistical information. The outcome of the analysis involves a large amount of statistical data on word frequencies in the texts (both absolute values and relative to other texts, or external corpora) and key words (words which have a high frequency in comparison with a reference corpus). A screen shot of the software is shown in Figure 5.1.

The software Concordance developed by [Wat09] has been created for people who need in-depth language or text analysis. It provides a free trial for the user. Concordance is able to generate indices and word lists, count word frequencies, compare different usages of a word, analyse keywords and publish the analysis results on the web. A screen shot of the software is shown in Figure 5.2

5.4 Exploratory Specification

Users may ask the following of a digital resource for the analysis of "Shakespeare in translation":

- Where, when, into which languages, and into which kinds of text has *Othello* been translated?
- How have translations influenced one another?
- How do versions vary in general?
- How do versions vary in particular?

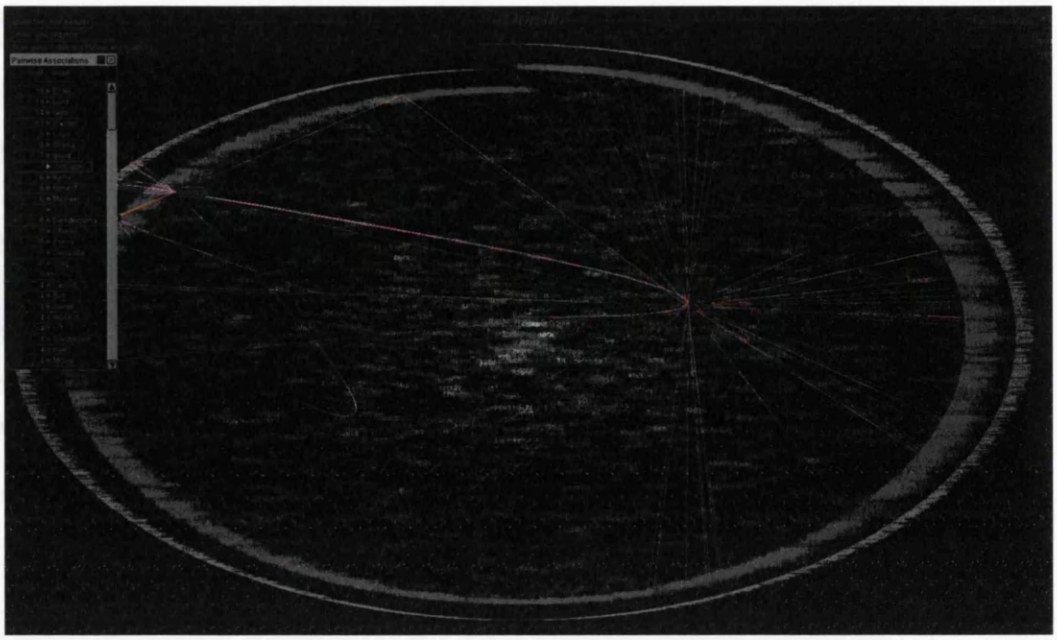


Figure 5.3: This figure shows the TextArc([Pal02]) visualization of Shakepeare's *Othello* in English. The entire text is depicted as an ellipse. Each line is drawn on the outside of the ellipse.

- How do translations deal with any specific terms or phrases that domain experts are interested in?

5.5 State-of-the-art Text Visualization

In this section, we investigate state-of-the-art text visualizations from two perspectives: research prototypes for text visualization and free off-the-shelf visualization tools. We refer to works by [Hom11], [RVA04] and [B10] for some lists of available text visualization software.

5.5.1 Research Prototypes for Text Visualization

This section introduces a list of research prototype text visualization techniques. Since 2005, we have observed a rapid increase in the number of text visualization prototypes being developed. As a result, various visual representations for text streams and documents have been proposed to effectively present and explore text features. In this section, we present some interesting and novel text visualizations which are able to present some of the extracted text attributes.

The ThemeRiver visualization, proposed by [HHWN02], depicts thematic variations over time within a large collection of documents. Thematic changes are shown in the context of a

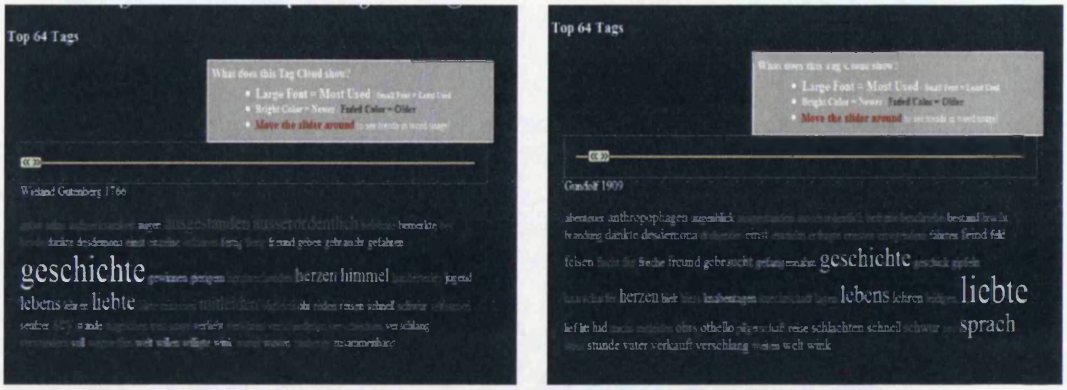


Figure 5.4: This figure shows the visualizations of two German translations of *Othello* using Tagline Generator([Meh06]). By moving the scroll bar, the user is able to see the visualization of each individual document. We experimented with more than 20 German translations of *Othello* using this tool.

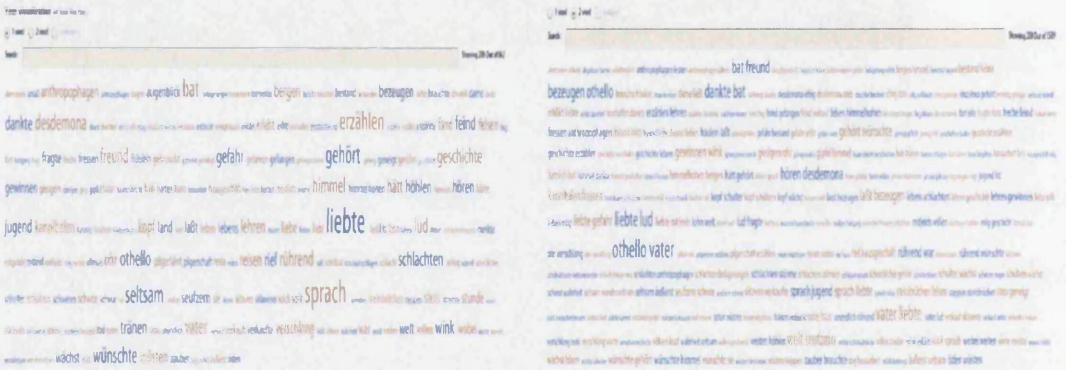


Figure 5.5: This figure shows the Tag Clouds([BGN08]) of *Othello* using ManyEyes([VWvH+07]). The left image depicts the tag clouds for every word, whereas the right image shows the tag clouds of pairs of words starting with letter "b". ManyEyes does not provide a text preprocessing option for the tag cloud, such as removing common words.

time line and corresponding external events. The focus on temporal thematic change within a context framework allows a user to discern patterns that suggest relationships or trends. A Document Contrast Diagram, proposed by [Cla08], is a visual summary of the content of two text documents that illustrates shared words, words that are unique to one document or the other, word frequency, relative size of the two documents, distribution of emotional tone within the documents, related words based on co-occurrence, and the most common word in each document segment. It uses the familiar bubble technique and effective use of color to contrast topic usage in two bodies of text. Parallel Tag Clouds, proposed by [CBW09], combines parallel coordinates and tag clouds to provide a rich overview of a document collection. Each

Figure 5.6: This image shows the Word Tree([WB08]) of *Othello* data using ManyEyes([VWvH+07]). When we input the word "liebte", all sentences containing this word are shown. The size of a word represents its frequency.

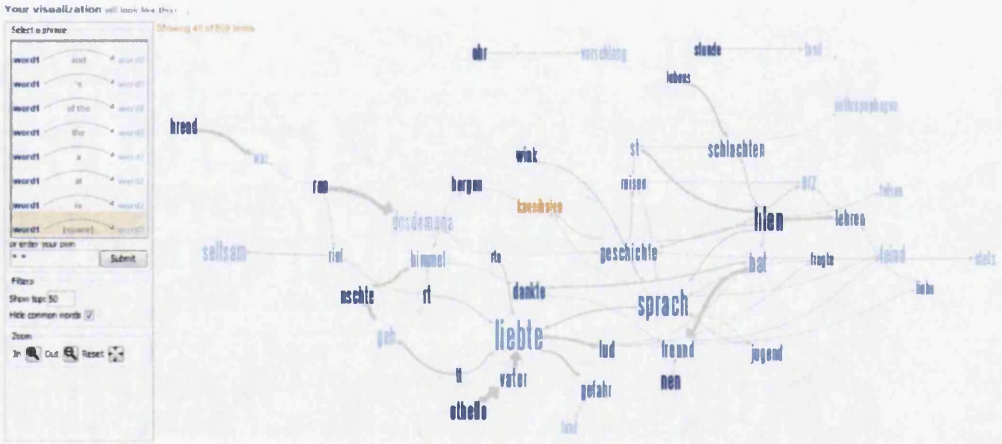


Figure 5.7: This image shows the Phrase Net([vHWV09]) of our *Othello* data using ManyEyes([VWvH+07]). It depicts any two words connected with open space in the *Othello* play. The size of the words depicts word frequency.

vertical axis represents a category. The words in each category are summarized in the form of tag clouds along the vertical axis. When clicking on a word, the same word appearing in other vertical axes is connected. Several filters can be defined to reduce the amount of text displayed in each category. This could help create more screen space and improve the clarity of the visualization.

DocuBurst proposed by [CCP09] uses a radial, space-filling layout to depict document content by visualizing the structured text. The structured text in this visualization refers to the

Figure 5.8: This image shows the TagCrowd([Ste08]) visualization of a passage from *Othello*. The common German words or stop lists are manually defined and removed from the original text.

Figure 5.9: This image shows the Wordle visualization([Jon09]) of a passage from *Othello*. The common German words have been removed.

is-kind-of or is-type-of relationship. For example, robin and redbreast is a kind of a bird. A bird is a kind of animal. An animal is a kind of organism or living thing. A living thing is an entity. As we can see, such structured text can form a tree hierarchy, with the entity as the root and robin or redbreast as the leaf. The root node of DocBurst visualization is shown as a circle. All other nodes are assigned to a sector of an annulus. The angular width of each sector is mapped to the number of leaves or children.

SparkClouds proposed by [LRKC10] integrates sparklines into a tag cloud to convey trends between multiple tag clouds. The sparklines can be used to present trends over time. From a

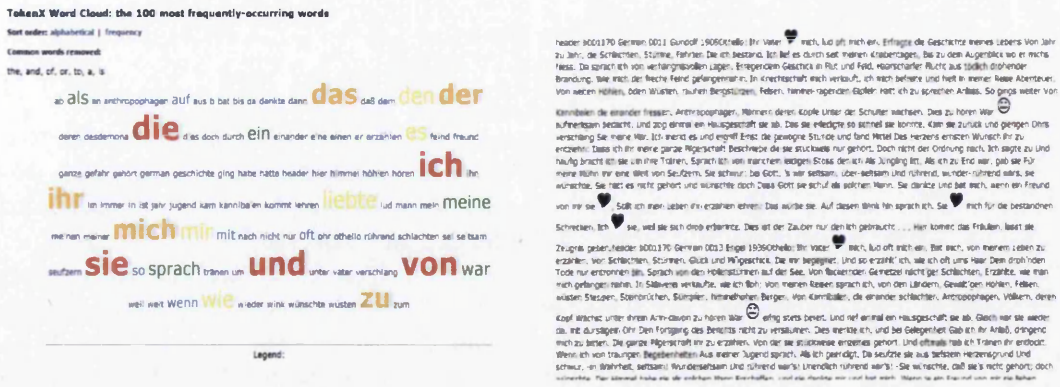


Figure 5.10: On the left image shows the Tag Cloud generated by ToxenX. The right image shows the text with the words "Liebte" replaced with a heart shape.

controlled study that compares SparkClouds with traditional trend visualizations, such as multiple line graphs, stacked bar charts and Parallel Tag Clouds. Results show that SparkClouds is more effective at showing trends along time. ManiWordle proposed by [KLKS10] provides flexible control such that a user can directly manipulate the original Wordle to change the layout and color. TextArc([Pal02]) is a visual representation of the entire text on a single page. It is an advanced combination of an index, concordance, and summary of the text. Animation is provided to enable the user to keep track of the variations in relationships between different words, phrases and sentences. In TextArc, the entire text is depicted as an ellipse. Each line is drawn on the outside of the ellipse. It preserves the typographic structure of the text. A word appears in the middle of an ellipse. A word with high frequency is displayed in brighter color and larger size. If a word is used more than once, it appears at the center of all of its mentions. The accepted data for TextArc is only from the TextArc library. Figure 5.3 shows the visualization of Shakepeare's play *Othello* generated by TextArc. NameVoyager([Wat05]) as a web-based visualization of historical trends in baby naming, has proven remarkably popular. The method used to visualize the data is straightforward: given a set of name popularity time series, a set of stacked graphs is produced. However, this tool does not accept user customized data sets.

5.5.2 Free, Off-the-shelf Text Visualization Tools

In this section, we investigate text visualization tools which are free to the public. Our work can facilitate modern language experts search for visualizations that benefit most the analysis of their collected Shakespeare translations. In this section, we experiment with these freely available tools on 23 German translations of *Othello*'s speech to the senate, *Othello*, act 1, scene 3. The following list summarizes the tools and visualizations we have tried:

- The ManyEyes([VWvH+07]) application with the following visualizations: Wordle, Tag Cloud, Phrase Net, Word Tree

- Tagline Generator([Meh06]): Tag Cloud, Time Line
- TokenX([Zil11]): Word Cloud, Word Highlighter

Tagline Generator [Meh06] is a simple PHP codebase that lets the user generate chronological tag clouds from simple text data sources without manually tagging the data entries. Once the users have populated the data source and configured the generator, it creates a list of all the unique words that have been used and counts their frequency. Next it identifies the different variations of words and combines them under the most common variation using the Porter Stemming Algorithm. The size of a word indicates its frequency in the document. The brightness indicates the year of the document, a newer document is brighter. The accepted data format of Tagline Generator is an XML file. Figure 5.4 shows the TagLine visualization of 23 German translations of Shakespeare's play, *Othello*.

ManyEyes [VWvH+07] is a free website where anyone can upload, visualize, and discuss data. It is an experiment created by the Visual Communication Lab. The input data of ManyEyes is not obtained by files, instead it accepts any forms of free text copied and pasted from any sources. It provides a number of text visualizations, such as Tag Clouds, Phrase Net and Word Tree. Again, we apply our *Othello* data, which contains 23 German translations of the play, to the visualizations in this tool. The standard Tag Clouds [BGN08] is a popular text visualization for depicting term frequencies. Tags are usually single words and are normally listed alphabetically, and the importance of each tag is shown with font size or color, as shown in Figure 5.5. There is also other software supporting Tag Cloud, such as TagCrowd [Ste08]. The advantage of TagCrowd over ManyEyes is that user can define the common words themselves and these common words will be automatically removed from the original text. The accepted input text is the same as for ManyEyes. Figure 5.8 shows the Tag Cloud visualization of our Othello data sets. The common German words are removed. Wordles are more artistically arranged (and often vibrantly colored) versions of a text. They tend to be less directly insightful as visualization, but often give a more personal feel to a document. The clouds give greater prominence to words that appear more frequently in the source text. Users can tweak their clouds with different fonts, layouts, and color schemes. Figure 5.9 shows the wordle visualization of some passages from *Othello*. The most commonly used German words which are of no interest are removed. Other software supports Wordle includes Wordlenet [Jon09], which is a tool for generating "word clouds" from text that the user provides. The accepted text format is the same as ManyEyes. Word Tree [WB08] is a graphical version of the traditional keyword-in-context method, and enables rapid querying and exploration of bodies of text, as shown in Figure 5.6. It is a visual search tool for unstructured text, such as a book, article, speech or poem. It allows the user to choose a word or phrase and shows all the different contexts in which the word or phrase appears. The contexts are arranged in a tree-like branching structure to reveal recurrent themes and phrases. The size of a word represents its frequency. Phrase Nets [vHWV09] illustrates the relationships between different words used in a text. It uses a simple form of pattern matching to provide multiple views of the concepts contained in a book, speech, or poem. Such as given a network of words and connection pattern word "and", where two words are connected if they appear together in a phrase of the form "X and Y", as shown in Figure 5.7.

Visual Mapping / Text Attribute		Word Frequency	Text Order Preservation	Document Trend	Document Relation
Value Encoding	Size	Doc Contrast Diagram [2008]			
		Tag Cloud [2008], Wordle [2009]			
		Phrase Net [2009]			
		Parallel Tag Cloud [2009]			
		Mani Wordle [2010]			
		Spark Cloud [2010]			
	Orientation	Wordle [2009], Mani Wordle [2010]			Doc Contrast Diagram [2008]
	Shape			ThemeRiver [2002]	
				NameVoyager[2005]	
				Spark Cloud [2010]	
	Color			ThemeRiver [2002]	
				NameVoyager[2005]	
Relation Encoding	Lines/Curves		Text Arc [2002]		Parallel Tag Clouds[2009]
			Phrase Net [2009]		
	Tree		Word Tree[2008]		
			DocuBurst[2009]		

Figure 5.11: This table is a classification matrix where the columns represent the visual mapping elements and the rows the text attributes. Each element of the matrix represents a visualization technique we have introduced in this chapter.

ToxenX created by [Zil11], is a powerful text analysis, visualization, and exploratory tool that has been customized for use on the Walt Whitman Archive. The text base for the Archive customization currently includes the six American editions of Leaves of Grass published in Whitman's lifetime and the deathbed edition of 1891-1892. TokenX currently supports the following features: text highlighting based on patterns in words, keyword in context, replacing words with blocks, word concordances sorted alphabetically or by frequency, word usage statistics, word substitution, user-selected replacement of words with images, creative exploration. The accepted input data format is same with Tagline Generator, they all accept the web XML file. Figure 5.10 shows two visualizations of *Othello* generated by TokenX.

5.6 Comparison

Our goal in this chapter is not a general comparison of information visualization tools, but rather a specialized comparison dedicated solely to the investigation of *Othello* text data. In this section, we conduct two types of comparisons dedicated to research prototypes and freely available software respectively. A comparison of the research prototypes is based on various text attributes each technique can afford and the corresponding visual design it adopts, whereas a comparison of the freely available software is based on the interaction supports for the user.

5.6.1 Comparison of Research Prototypes

We have generated a classification matrix table for the comparison of research prototypes. As an element of the matrix, each text visualization technique can be assigned according to the type of text attribute it depicts and the adopted visual element mapped to that attribute. The text attributes can be categorized into four parts including word frequency, word ordering in a

context, document trend over time and document relationship. The word frequency refers to the number of times an individual word appears in a document. The word ordering indicates the order or sequence in which a word appears in a sentence, paragraph, document or other contexts. The document trend implies how the topic, theme or other properties of a document evolve over time. The document relation describes how different documents correlate with one another in respect to the common properties they share. Besides the text attributes, we also categorize different types of visual elements adopted in our investigated text visualization techniques. According to the classification proposed by [Spe01], we are able to categorize the visual elements by their encoded data types. A value encoding maps a visual element to a single data value. The properties of a visual element for such encoding includes the size, orientation, shape and color. A relation encoding aims to depict the relations between multiple data values, whose visual elements include the lines or curves connecting various data items and tree depicting the hierarchy. Shown in Figure 5.11 is a summary table for our proposed classification matrix. As we can observe from this table, the word frequency is represented by the size of a visual element in most of the visualizations. In addition, Wordle and ManiWordle also utilize different orientations for a compact and aesthetic layout. The word ordering is mostly represented by the relation encoding, for example, Text Arc and Phrase Net utilize lines and curves to connect two subsequent words in different contexts. Moreover, a word ordering can be presented in a hierarchical manner such as with Word Tree and DocuBurst. In order to visualize document trends, three methods can be adopted as shown in the table. These methods have applied shape and color in a stack graph to visualize the document evolution over time. From the last column of the table, we can see there are two methods available, namely Parallel Tag Clouds and Document Contrast Diagram, to visualize the relationship across multiple documents. Parallel Tag Clouds is advantageous over Document Contrast Diagram in respect to the scalability to a large document corpus. However, one disadvantage of this visualization is its incapability to display groups of words which are missing in one document but frequently appear in the others. When we explore the variations among the *Othello* translations, the domain experts would like to know groups of words which a particular author never uses but which frequently appear in other authors' work. Also, brushing multiple words in different documents might introduce clutter due to crossing lines in parallel tag clouds. In order to address this problem, we have proposed a new method to represent multiple documents for *Othello* translations in Section 5.7,

Tool \ Interaction		Color	Layout	Search	Filtering	Context	Zoom	Timeline
ManyEyes	Tag Clouds			✓		✓		
	Wordle	✓	✓		✓			
	Phrase Net		✓	✓	✓		✓	✓
	Word Tree		✓	✓		✓	✓	
TokenX	Word Cloud			✓	✓			
TagLine Generator								✓

Figure 5.12: This table summarizes the interaction designs for different freely available text visualization tools.

5.6.2 Comparison of Freely Available Tools

In Section 5.5.2, we have discussed three freely available text visualization software. They provide a variety of visualizations and integrate with different interaction options. These interaction paradigms of each tool are systematically applied to every visualization component within that tool. In this section, we aim to compare and evaluate the freely available text visualization tools from the perspective of interaction design. As shown in Figure 5.12, several different interaction options are listed including color mapping, layout manipulation, keyword searching, common word filtering, context view, zooming and time line. Color mapping allows the user to change different color scales for the visual elements of a visualization. We find that only Wordle in ManyEyes supports this user option. Layout is another important property in some of the visualizations, such as Wordle, Phrase Net and Word Tree. ManyEyes offers different layout styles for the user in these visualizations. Keyword searching enables the users to quickly obtain and retrieve the terms in which they are interested. From the summary table, most of the tools have offered searching functions except for Wordle and Tagline Generator. We note that the searching criteria varies in different visualizations, for example, Tag Clouds allows word searching, Phrase Net supports word relation searching and Word Tree enables sentence searching. The filtering process allows the user to define and remove the most common words from the original text. This is important in our *Othello* translation analysis as we have to remove the words which are of no interest to the domain experts. We note that common word removal can also be done in text preprocessing. A context view displays a sentence or paragraph in which a given keyword appears. Zooming enables drill down into different parts of a visualization for more details. The time line enables the user to freely switch between different document views along the time series. As we can observe from the summary table, visualizations in ManyEyes have offered the most interaction support. However, it cannot support multiple documents comparison. The only tool which supports multiple document comparison is Tagline Generator. Therefore this tool is of particular interest to the domain experts.

5.7 Proposed Visualization

In this section, we briefly introduce our proposed, customized visualization enabling users to explore how multiple different translations of *Othello* relate to the base text (that is, the established, curated English text of Shakespeare). Our proposed visualization consists of two parts: the first part is designed for macro analysis using parallel coordinates which depicts an overview of the abstract information from different documents, while the second part is designed for micro analysis which allows the user to zoom into the base document and segments for in-depth reading.

5.7.1 Macro View: High-level Text Analysis

Parallel coordinates introduced by [Ins09] is a widely used visualization technique for exploring large, multidimensional data sets. It is powerful in revealing a wide range of data characteristics such as different data distributions and functional dependencies as stated. The

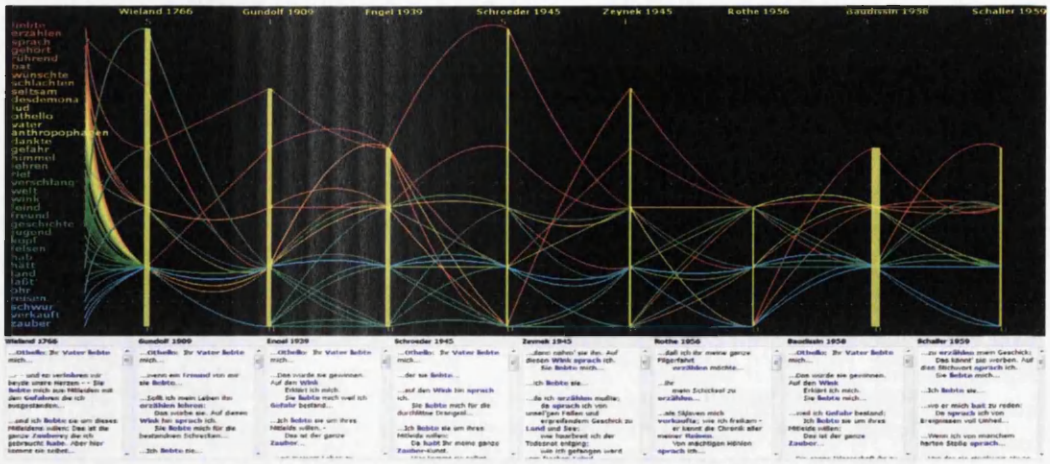


Figure 5.13: This image shows an overview of our visualization. The parallel coordinates illustrates a focus view of the term frequency. The text boxes below the parallel coordinates show the context views. They present the entire sentences from the original text where each keyword appears.([GLC+11])

textual information of each document can be transformed into a vector. In our parallel coordinates, we encode the document dimensions as term frequencies. Domain experts from Arts and Humanities selected eight interesting translations according to their similarity score. For the initial analysis, we chose a significant passage from the play, Othello's pivotal speech to the Venetian Senate in Act1, Scene3: the longest single speech in the play (about 300 words in Shakespeare's text). Figure 5.13 shows an overview of our visualization. The detail of this visualization system is introduced in Chapter 6.

5.7.2 Micro View: In-depth Text Analysis

We know that the stability or similarity of literary translations varies, not only according to who is the translator and the circumstances of their work, but also depending on the specific problems of translation and interpretation which are posed by specific parts of the text. It is of interest to discover how differences among multiple translations are associated with particular parts of the translated base text. This demands a visualization which permits the user to navigate and read the full text. In this model, the base text of *Othello* is divided into contiguous segments. Normally a segment is a speech (usually between one and one hundred words, and occasionally longer), or a sentence within a speech. Following text preparation, all the translations are aligned with the base text, segment by segment: a machine-assisted manual procedure. Then, for each set of translations of each segment, a concordance is generated. Based on this segment concordance, I have developed a coordinated-multiple-views visualization system to present, analyze and explore segment variations of German translations of *Othello*. This visualization system is introduced in Chapter 7.

Chapter 6

Visualizing Translation Variation on Term Level: Shakespeare's *Othello*

Contents

This chapter is organized as follows: In Section 6.1, we describe our source data. In Section 6.2, we explain how are the original documents processed before being input to the visualization. In Section 6.3, we illustrate our structure-aware Treemap for meta data analysis. In Section 6.4, we present the Focus + Context parallel coordinates for translation variation exploration. In Section 6.5, we report the feedback from the domain experts.

6.1 Background Data Description

The domain experts from Arts and Humanities have collected 57 different German translations of Shakespeare's play, *Othello*. For each translation, metadata recorded includes the author name, publication date, country, title of the play and impact index. The translations were written between 1766 and 2006 in seven different countries defined including Germany (pre-1949), East Germany (1949-1989), West Germany (1949-1989), FRG (Germany since 1989), Austria, Switzerland and England. The impact index refers to each translator's productivity and reputation. it includes the re-publication figures or each *Othello* translation. Figures were derived from the standard bibliography of Shakespeare in German [HS03]. The index has five levels ranging from 1 to 5, where 1 means that the translator is not listed in the bibliography and 5 means that more than 50 publications and re-publications by the translator are listed in the bibliography. Figure 6.1 shows the chronological distribution of our collected documents.

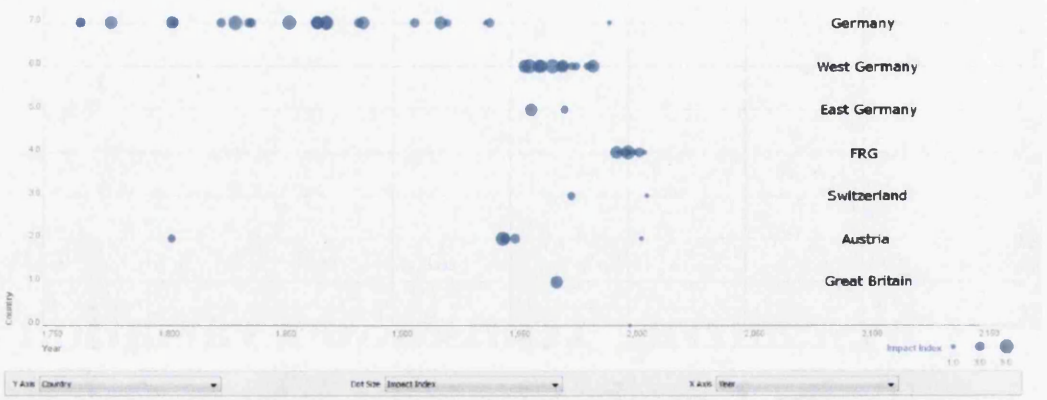


Figure 6.1: This image illustrates the distribution of our collected German Othello translations. The X-axis is mapped to the publication date and Y-axis to seven different countries. The dot size is mapped to the impact index. A larger radius depicts a translation with higher republishing figures.

The X-axis is mapped to the publication date and Y-axis to the different countries. The ellipse radius is mapped to impact index.

6.2 Text Preprocessing

Before the original translation can be analyzed within our visualizations, we need to generate various features from the textual information and transform them into numerical vectors. In this work, we process our original text in five steps, namely document standardization, tokenization, stemming, vector generation and similarity calculation. The major outputs include making concordance of each document and computing their similarity.

Since the *Othello* translations are collected from various sources (some PDF, some archival typescripts, mostly books), we firstly transform and integrate them into a standard XML format. Next, document tokenization breaks the stream of text into a list of individual words or tokens. During this process, common words carrying little meaning which are not of interest to domain experts, such as "der" (the), "da" (that) etc, are eliminated from the token list. Furthermore, stemming reduces all of the tokens to their root forms. Based on this cleaned and standardized token list, we are able to generate a concordance table for each document by counting the frequency of every unique token.

For in-depth document comparison, we also need an objective document similarity measure. The domain experts from Arts and Humanities suggest a list of high-frequency keywords as a search query. This keyword list can be extracted from multiple interesting documents. The similarity between our collected translations can then be measured using the LSI (Latent Semantic Index) model [DDF+90]. This model is widely used in information retrieval where the list of terms associated with their weight is treated as the document vectors. The weight of each term indicates its importance in a document, and is given by $Tf \times Idf$. We use Tf (Term

Frequency) to refer to the number of times a term occurs in a given document, which measures the importance of a word in a given document. *Idf* (Inverse Document Frequency), as its name implies, is the inverse of the Document Frequency. The Document Frequency is the number of documents in which a word occurs within the collection of documents.

Thus the weight of a term i in document j can be defined as:

$$w_{i,j} = tf_{i,j} \times idf_j = tf_{i,j} \times log\frac{N}{df_i}$$

where N is the total number of documents in the corpus, *df* is the document frequency and *idf* is the inverse document frequency. Large values of $w_{i,j}$ imply term i is an important word in document j but not common in all documents N.

Then a document j can be represented as a vector with each dimension replaced by the term weight:

$$\vec{D}_j = (w(0,j), w(1,j), ..., w(n,j))^{\mathbf{T}}$$

A large number of words in the search query might lead to an extremely high-dimensional document vector, so we use SVD (Singular Vector Decomposition) to perform a dimension reduction. Then the similarity between the two documents j and k can be measured by the angle between these two vectors:

$$\cos\ Sim(D_j, D_k) = \frac{\vec{D}_j \cdot \vec{D}_k}{|\vec{D}_j||\vec{D}_k|}$$

Such similarity measures are generated for all of our *Othello* translations. This information is featured in our treemap and parallel coordinates.

6.3 Structure-aware Treemap

As discussed in Section 6.1, metadata of each document includes author name, play title, date, place of publication and impact index. The scatterplot in Figure 6.1 is able to present the overall historical distribution, but it cannot provide an aggregation of the data. For example, if the user wants to explore or rank the total number of translations, or the total number of re-publications in any century, decade or country in our document collection, the scatterplot is unable to convey an answer. Next to this, we observe that the meta data can be arranged in a hierarchical structure. For example, each century breaks down into several decades. In each decade a few translations are published in several countries. In each country several authors published their work. For each author his translations have the impact index. Given this structure, we are able to generate a Treemap [JS91, Shn92] visualization.

The traditional treemap is able to compare the node values in any tree level. But it lacks the ability to show the entire tree structure intuitively. For tracing the treemap hierarchy, it's necessary to only list the relevant substructure which shows the ancestor and descendants of the interested node. The Degree-of-Interest tree [CN02] provides a clear hierarchy at a low cost of screen space by changing the viewpoint and filtering out the uninteresting tree nodes. In addition, it offers instant readability of the node labels. Therefore, we adopt linked views

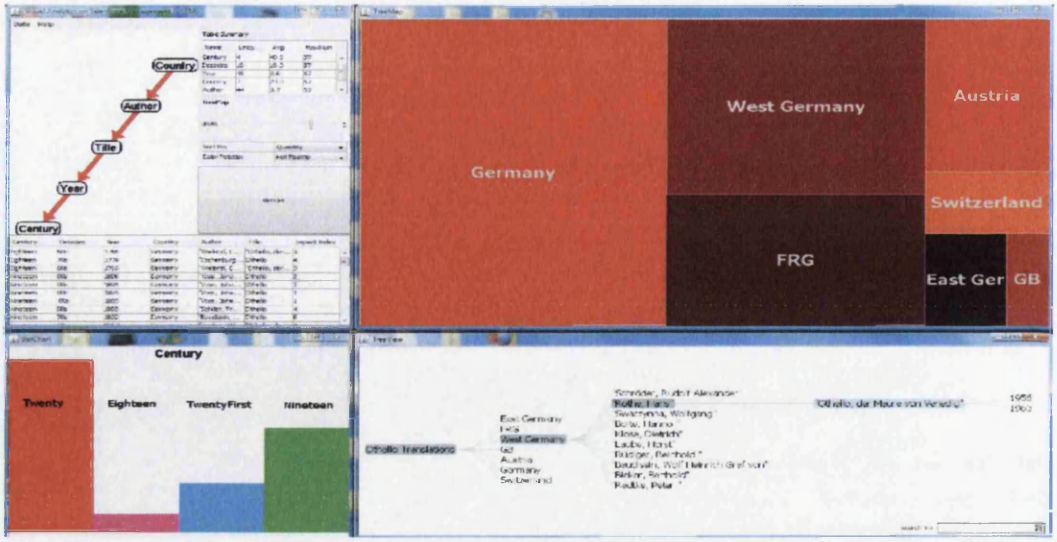


Figure 6.2: This image illustrates the interface of our structure-aware treemap. The left part shows the control panel by which the user is able to manipulate the tree hierarchy, compare the values in each hierarchy via a bar chart and set up the configuration for the visualization. Also the user is able to select their interesting documents from the spreadsheet. The right part shows the treemap and DOI-tree. The area of the leaf node is mapped to the quantity. As we drill down and up to different tree levels, the DOI-Tree keeps track of the structure. Also, the DOI-tree could initiate a searching task.

using both DOI tree and treemap to enable structure tracing. Our system is composed of two parts, namely the control panel and structure-aware treemap. The control panel is shown on the left half of Figure 6.2. It extracts the ontological hierarchy information from the input data sets and sets up the configuration for the visualization. The user is able to change the order of hierarchy or reduce the number of hierarchies by moving the graph nodes. The right half of Figure 6.2 is a structure-aware hierarchical visualization, containing the coordinated views of the squarified treemap and DOI tree [CN02]. As we traverse back and forth between the intermediate levels of the treemap, the DOI tree view clearly keeps track of how each selected node is derived from its ancestors.

The area of the leaf node can be either mapped to the impact index, the similarity measure or the quantity. In Figure 6.2, from the bar chart, we learn that most of our collected translations were published in the twentieth century. During this century, most translations are published in the 1940s and 1970s. For the domain specialists, this raises questions about possible correlations with comparable datasets (translations of other or all Shakespeare plays), and about possible correlations between periods in German history, and specific interest in Othello. By changing the hierarchy, we also learn that although the documents are all translations of *Othello*, they have different titles: the commonest titles of the translations are "Othello" or "Othello, der Mohr von Venedig", some authors use the title "Die Tragedie von Othello, dem

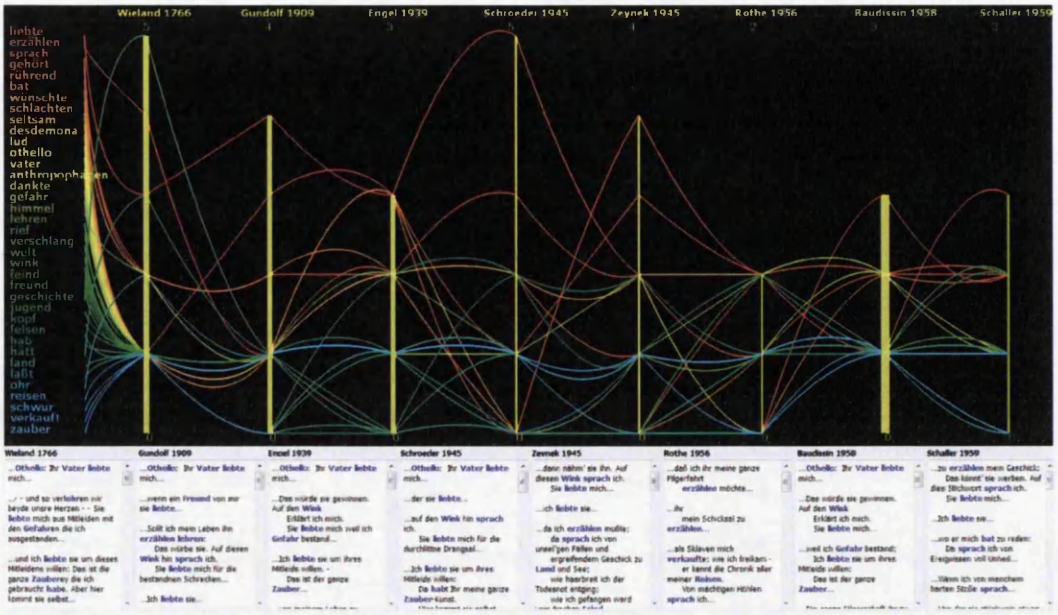


Figure 6.3: This image shows an overview of our visualization. The parallel coordinates illustrates a focus view of the term frequency. The text boxes below the parallel coordinates show the context views. They present the entire sentences from the original text where each keyword appears.

Mohren von Venedig", two use "Othello, der Maure von Venedig" and one author uses the title "Othello, Venedigs Neger". These outliers are of particular interest to the domain experts.

Our treemap system helps users manage their documents, such as ranking the documents according to different criteria, analyzing the global features of the metadata and selecting the interesting documents. It can be scaled up to include new datasets such as translations of other works by Shakespeare and enable users to explore common patterns in the metadata. The DOI tree can initiate the searching task by which a user is able to search terms in any hierarchy. Since the collection of the German translations is still expanding by domain experts, our treemap will play an increasingly important role in the meta data analysis.

6.4 Focus+Context Parallel Coordinates

Parallel coordinates, introduced by Inselberg and Dimsdale [ID90b, Ins09] is a widely used visualization technique for exploring large, multidimensional data sets. It is powerful in revealing a wide range of data characteristics such as different data distributions and functional dependencies [KK96]. As discussed in Section 6.2, the textual information of each document can be transformed into a vector. In our parallel coordinates, we encode the document dimensions as term frequencies.

Domain experts from Arts and Humanities selected eight interesting translations according

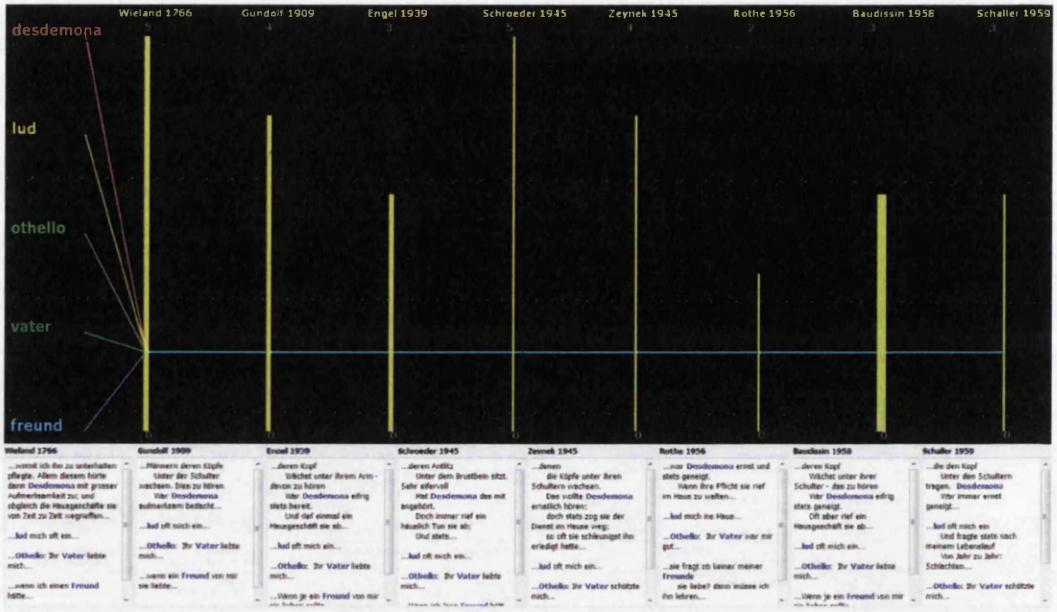


Figure 6.4: In this image, we obtain five keywords which only appear once in all documents.

to their similarity score. For initial analysis, we chose a significant passage from the play, Othello's big speech to the Venetian Senate in Act1, Scene3: the longest single speech in the play (about 300 words in Shakespeare's text). Figure 6.3 shows an overview of our visualization. The column on the far left displays a list of selected keywords: these are most frequently occurring significant words in the document corpus. The parallel coordinates present a focused view of keyword frequencies. Each document is represented by a vertical axis. In order to maintain a unified scale, the height of each vertical axis is made proportional to the range between each document's minimal and maximal word frequencies. Zero frequency simply means that a keyword has not occurred in that document. The thickness of each vertical axis is mapped to the document's similarity with others in terms of LSI score: a thicker line means a higher similarity value. The number of occurrences of each keyword in each document is connected by a polyline. Each polyline is rendered in a different color to enable visual discrimination. The text boxes below the parallel coordinates provide context views for keywords selected by the user. Each text box represents an individual document and shows the entire sentences from the original text where each selected keyword occurs. We also apply edge bundling to enhance the visual clustering and the user is able to control the curvature of all the edges [ZYQ+08]. Curves with the least curvature become a straight line.

We provide various interaction support, such as selection, brushing and linking. As the user selects individual or multiple keywords, the corresponding polylines are rendered. The user can also select various frequency levels in any document and the corresponding keywords having that frequency are displayed. Along with the selection and brushing, the text boxes which show the context views keep updating.

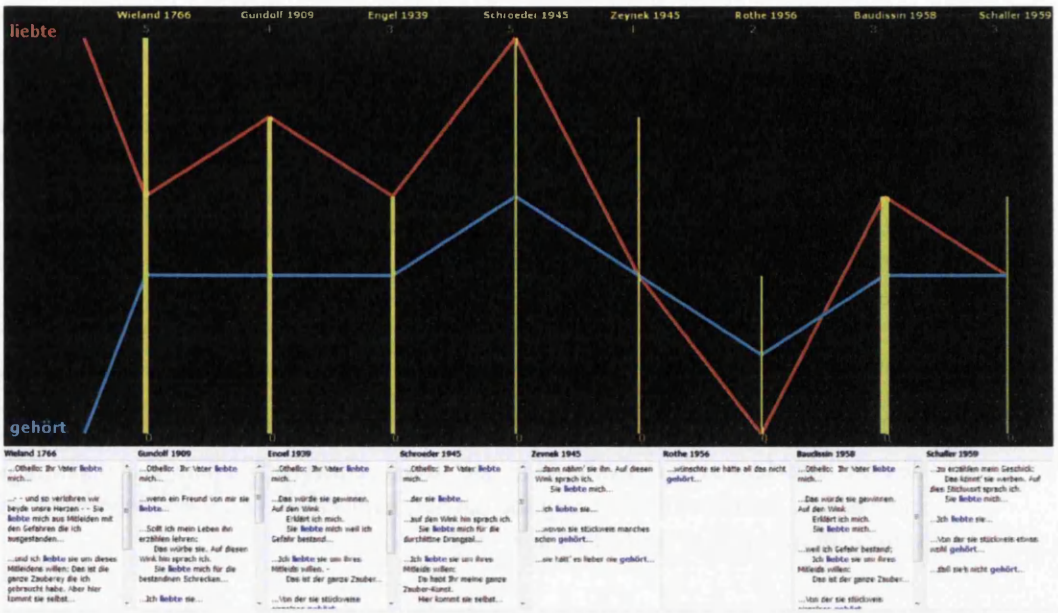


Figure 6.5: In this image, there are two keywords showing a strong correlation.

Our system also supports composite brushing such as an AND-bush or OR-bush [HLD02b]. We can use the AND-Brush to obtain all keywords which occur in every document: words used by all translators regardless of the translators' reputations and impact. If we brush the keywords which do not appear in document "Baudissin 1958", we learn that this document contains all the keywords except "fand". This helps to explain why this document has the highest similarity score. The domain experts indicates that this finding is surprising and interesting. As shown in Figure 6.4, we observe five keywords which appear just once in all the documents. From the context views, the sentences containing these two words are almost the same in every translation. As shown in Figure 6.5, there are two keywords showing a strong correlation. Both findings raise interesting questions for the domain experts.

6.5 Domain Expert Reviews

This section is written by our domain expert, Tom Cheesman, who have collected all of the German translations. The focus+context parallel coordinates permits comparative visualization and exploration of concordances. A concordance is normally displayed as a simple list of words in a vertical column (in order of frequency or alphabetically). Standard concordance software also offers the option to display contexts of use for a particular word (i.e. the different word strings in which a word appears). This tool successfully combines a concordance-derived keyword list and context views with display of frequencies of words across multiple, comparable versions, in the form of parallel coordinates. This is a promising way of exploring texts through their different uses of meaningful words. In the display of parallel coordinates, the

composite brushing enables us filter for any correlations between word-uses, positive or negative: pairs/groups of words which appear together, or never appear together. The similarity of each document tells us an objective measure of how similar each document is to the keyword lists. In this particular case, the visualization tells us that Baudissin's translation-which is the standard, most often republished and performed German translation of the play - contains the most keywords in this speech which are common to most of the other translations. Since other translations are produced and marketed as "alternatives" to Baudissin, this high degree of apparent dependency on the standard translation is surprising, and it demands further investigation.

Our current corpus of German *Othello* translations is relatively small (under 60 documents), but we envisage it growing: in respect of other works (Shakespeare's many other plays, and poems; and potentially works by other writers) and also in respect of other languages of translation (at least one of Shakespeare's works exists in about 100 languages). Hence, the flexible metadata overview offered by the structure-aware Treemap visualization will become increasingly valuable in managing the dataset, exploring its various dimensions and selecting subsets of translations for further analysis.

Chapter 7

Visualizing Translation Variation on Segment Level: Shakespeare's *Othello* : ShakerVis

Contents

7.1 INTRODUCTION

William Shakespeare's plays have been translated into every major living language. In some languages, his plays have been re-translated many times. These translations and re-translations have been produced for about 250 years, in varying formats: some as books, including reading editions and study editions; some as scripts for performances (theatre, film, radio and television scripts). Multiple heritage text translations have remained, until now, an untapped resource for Digital Humanities. Divergence of multiple kinds caused by various factors is normal among multiple translations, due to differing translation purposes, genetic relations (translators 'borrowing' from one another), context-specific ideological and cultural influences, inter-translator rivalry, and translator competence and style. Studying variations in re-translations of world cultural heritage texts is of cross-cultural interest for humanities researchers. This does not just apply to Shakespeare. Variations among re-translations reveal histories of language and culture, intercultural dynamics, and changing interpretations of every translated work.

Digital Humanities researchers working on a project called 'Translation Arrays: Version Variation Visualization', have collected an experimental corpus of fifty-five different German

re-translations of Shakespeare's play *Othello* (1604). The translations date from between 1766 and 2010. Most texts were acquired in non-digital formats. A representative sample of 32 of the re-translations has been digitized. The 32 texts of one scene of the play have been cleaned, formatting normalized, all texts segmented, speech by speech, and all segments semi-automatically aligned with a so-called 'base text' (Shakespeare in English), to create a parallel corpus. The selected scene is Act 1, Scene 3: in Shakespeare's original text. This scene is c. 10% of the play's length; it has c.3,000 words from the play's total of c.28,000 words; and the scene has 88 speeches. This parallel corpus can be accessed at the Translation Arrays project website: *www.delightedbeauty.org/vvv*. Based on this corpus, the team want to explore variations between different translations at the segment level, in order to uncover patterns relating to different types of translation, historical periods, genetic relations, and patterns relating to different sub-sets of segments. Sub-sets include speeches by certain characters (with the hypothesis that translators interpret characters in the play in distinctive ways, and therefore translate their speeches in different ways), and segments with certain linguistic and poetic features, such as metaphors, puns, rhyme, interpretative challenges, and so on. The team's general long-term aim is to develop analytic tools which will work for any corpus of re-translations. In this chapter, the domain experts have selected a subset of their collected translations which are of great interest and they would like to analyze and explore the variations between them. The detailed information of these selected documents is discussed in Section 7.3.

Based on this collection, we attempt to devise a statistical metric to compute the similarity coefficients between pairs of documents, i.e. translations or versions of each segment, on the basis of lexical concordances. The original textual information is converted to a term-document matrix and further projected onto a lower-dimensional space. These document vectors with reduced dimensionality can be presented, analyzed and explored by our novel, application-specific interactive focus+context visualization system. From our visualization, we are able to obtain an overview of the distributions and relationships between documents of various segments. By the means of interaction support, the user is able to explore the underlying clusters, outliers and trends in the document collection. A focus view enables in-depth comparison between documents in order to identify the textual details behind these patterns. In the end, we can identify which segments from the original play provoke very different translations and which are characterized by similar translations, i.e stable content. Our tool is evaluated by the domain experts who are studying this topic. The findings help them better understand how different German translations of *Othello* relate to one another and to the base text.

In this chapter, we contribute the following:

- We develop an interactive visualization system, abbreviated as ShakerVis, for presenting, analyzing and exploring segment variations between German translations of *Othello*.
- We derive statistical metrics, such as Eddy and Viv values to measure the stability of segment translations of *Othello*.
- Our system is evaluated by the domain experts. Some interesting patterns and findings are discovered.

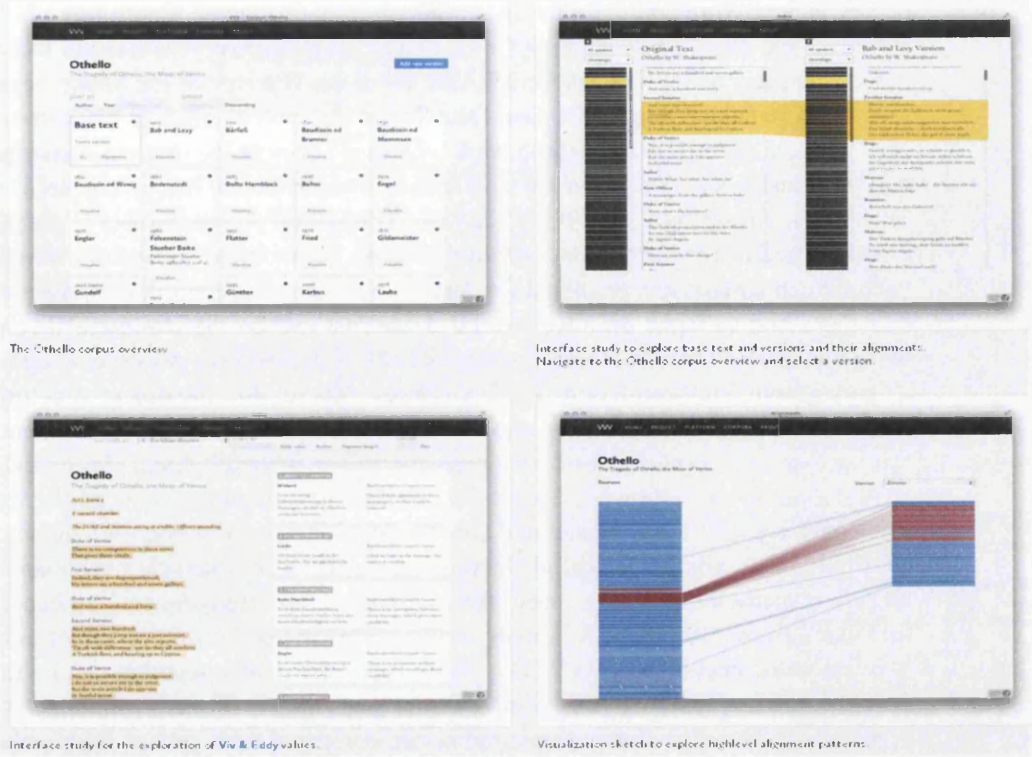


Figure 7.1: This image shows an overview of four interfaces of the Translation Arrays tool suite [CFT12].

The rest of the chapter is organized as follows: Section 7.2 discusses previous work related to our approach and the problem domain. Section 7.3 describes the specific group of *Othello* translations we are using in this chapter. Section 7.4 demonstrates the key ideas in preprocessing the textual data, projecting the data onto lower dimensional space and computing a similarity value for each segment translation. Section 7.5 presents our visualization and interactions to explore and analyze the derived document statistics. Section 7.6 reports the feedback from the domain experts who are studying this problem.

7.2 Related Work

Our previous approach tries to visualize how each unique term changes in each translation, whereas in this chapter we would like to work on a more abstract document level, namely segment or speech of German translations of *Othello*. Understanding which segments remain stable and which exhibit high variability sheds new light on the local culture with respect to both the time period and region. Therefore, our major goal for this project is to develop an interactive visualization system to present and explore the parallel segment variations between multiple translations. Stephan Thiel's work presents all the plays of Shakespeare, using the

deeply tagged WordHoard digital texts, filtered through analytic algorithms [Thi06]. DocuScope is a text analysis environment with a suite of interactive visualization tools for corpus-based rhetorical analysis [Car98]. Michael Witmore, Director of the Shakespeare Folger Library, and Jonathan Hope have used DocuScope for years to analyze Shakespeare and other early modern texts [HW04]. These work effectively present the original Shakespeare's work, but not translations. The previous work which is more related to this chapter is presented in Translation Arrays tool suite [CFT12]. The Translation Arrays project is creating tools for exploring and analyzing corpora of re-translations, i.e. multiple translations into the same language. Such corpora can be mined for data on the past and present development of translating languages and cultures, on inter-cultural dynamics, and on the interpretability of translated works and parts of works. Recently the project team created a corpus store, a segmentation and alignment tool, and web-based visual interfaces. These offer alignment structure overviews, navigation through parallel texts, and a comparison of two versions of a segment alongside a full base text view (with back-translations from German to English). An overview interface of these interfaces is shown in Figure 7.1. In the last mentioned view, all the translations of a selected segment are retrieved and can be sorted in several ways, e.g. author name, date, or length, or by relative lexical distinctiveness, or distance from other versions. We call this relative distance value 'Eddy', from the metaphor 'eddy' (turbulence) and because it can be calculated from concordances in many ways, all involving the sum of values associated with individual documents [CtVVVPT11]. Thus, all versions of a segment can be ranked in this view, in order of distinctiveness. In a further step, the set of Eddy values for versions of a segment can be reduced to a single value and compared with sets of Eddy values for other segments. This value is termed 'Viv' (vivacity). The base text is annotated with Viv in the website, so as to identify 'hotspots', where translations are most different. The work presented in this chapter develops a new metric for 'Eddy' and demonstrates visualizations which enable users to identify clusters and outliers in re-scalable text and segment corpora. Future work integrates these visualizations into the project's web-based tool suite, and devises a metric for aggregating these 'Eddy' results into a 'Viv' annotation.

7.3 Background Data Description

In this chapter, we concentrate on the visual analysis of parallel segment variation. A segment refers to a section within a document, of arbitrary size. Segments might be lexical terms, phrases, or sentences, in any text; or acts, scenes, and speeches in play-texts; or chapters, paragraphs, and spoken dialogue in works of prose fiction; or chapters and verses in works of scripture; and so on. In our current work, each speech in the play is regarded as a segment. Equivalent speeches in the German translations have been aligned with the English base text. Alignments can be problematic and complex, because some re-translations re-order and omit material from the base text and add new material with no base text equivalent. The experiment reported here uses a selected sub-corpus: ten re-translation texts of known interest, and seven parallel segments from each. The segments were selected for non-problematic alignments and for comparable, relatively high segment lengths (42 to 95 words in the base text). They consist of the seven consecutive longer speeches which begin in the base text with Desdemona's speech

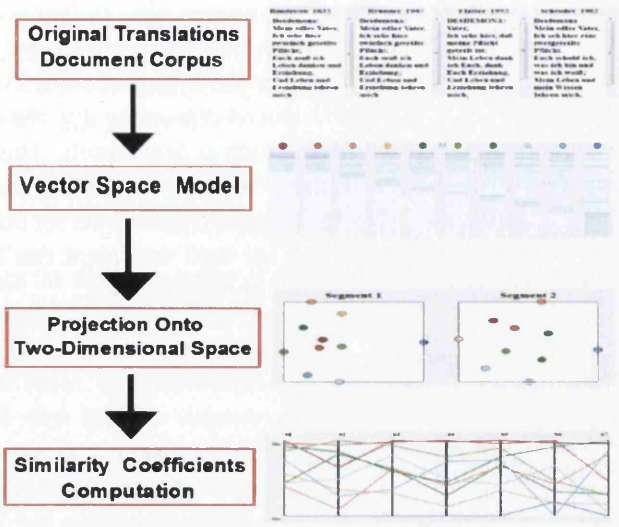


Figure 7.2: This diagram demonstrates how our statistical coefficients are derived and the way they can be visualized.

'My noble father' (excluding three very short speeches beginning with the Duke's speech 'If you please'). The ten re-translations investigated include: (a) two different editions of the standard verse translation for performance and reading (Baudissin 1832, as edited in 2000 for Project Gutenberg, and as edited by Brunner in 1947) [Bau32, Bru47]; (b) two didactic prose translations for students (Engler 1976, Bolte 1985) [Eng76, Bol85]; (c) one recent prose translation for performance (Zaimoglu 2003), known to be an outlier because the text is very idiosyncratic [Zai03]; and (d) five verse translations for performance, or for performance and reading, dating from the 1950s-1970s (Flatter 1952, Schröder 1962, Fried 1970, Lauterbach 1972, Laube 1977) [Fla09,Rud63,Fri99,Lau96,Lau78]. The genetic and stylistic inter-relations of these five versions have not yet been studied, but all are considered 'complete' and 'faithful'.

7.4 FUNDAMENTALS

In this section, we utilize statistics to measure the relative distinctiveness of a segment or document, in relation to other German translations. In order to achieve this, several steps are implemented, such as, converting the original text into vector space, reducing the document dimensionality and computing the average similarity value, as depicted in Figure 2.4. We initially pre-process the original document corpus which contains ten different German translations of *Othello*. Each translation contains seven speeches, namely segments. A segment in one translation is semi-automatically aligned to the same segment in the other translations. The text preprocessing transforms the original document into a term-document matrix. A document can then be regarded as a vector with each dimension representing a unique term, as discussed in Section 7.4.1. Because the derived document vector suffers from high dimension-

ality, it is noisy due to the existence of uninteresting instances of terms. Also visualizing and analyzing documents in such a high-dimensional space can be challenging. Therefore we utilize the multi-dimensional scaling technique to project original document vectors onto a lower dimensional space [Dav92]. With reduced dimensionality the document can be presented by conventional visualization techniques, such as scatterplots. This helps the domain expert visually identify and recognize the clusters, outliers and trends between documents, as discussed in Section 7.4.2. Finally, we compute similarity coefficients for documents in different segments. In addition, a global similarity value for each document can be obtained by calculating the diameter of each segment, as discussed in Section 7.4.3.

7.4.1 TEXT PRE-PROCESSING

During the text preprocessing, we process our original texts in six steps, namely document standardization, segmentation, alignment, exclusion of non-relevant text elements, and tokenization. Since the *Othello* translations are collected from various sources (some PDF, some archival typescripts, mostly books), we firstly transform and integrate them into a standard XML format. Next, we define contiguous segments for each document and align the segments with the English-language base text, using machine-supported manual methods. In this process we also define and exclude some components of the original text which we do not want to process: such as stage directions, editorial notes, and etc. However the names of speakers for each speech are provided in the output display. This leaves the text which is relevant for similarity calculation: the speeches. Then, tokenization breaks the stream of text into a list of individual words or tokens. During this process, we can also experiment with selecting certain words for inclusion or exclusion from the token list, such as common 'function words' or 'stop words' carrying little meaning; also with stemming, to remove suffixes, prefixes, and grammatical inflections; and with lemmatization, to reduce all tokens to their root forms. These techniques will be carried out in the future work. Based on this cleaned and standardized token list, we are able to generate a concordance table for each segment by deriving the frequencies of every unique token in every translation segment.

7.4.2 DIMENSION REDUCTION

After the original document has been cleaned and pre-processed, we are able to construct a weighted term-document matrix where the list of terms associated with their weight is treated as document vectors. The weight of each term indicates its importance in a document. Empirical studies report that the Log Entropy weighting functions work well, in practice, with many data sets [LMDK07]. We use Tf (Term frequency) to refer to the number of times a term occurs in a given document, which measures the importance of a word in a given document. We use Gf to refer to the total number of times a term i occurs in the whole collection. Thus the weight of a term i in document j can be defined as:

$$\omega_{i,j} = (1 + \sum_{j} \frac{\frac{tf_{i,j}}{gf_i} \log \frac{tf_{i,j}}{gf_i}}{\log n}) \log(tf_{i,j} + 1) \tag{7.1}$$

where n is the total number of documents in the corpus. The term gf_i is the total number of times a term i occurs in the whole collection. Large values of $\omega_{i,j}$ imply term i is an important word in document j but not common in all documents n.

Then a document j can be represented as a vector with each dimension replaced by the term weight:

$$\vec{D}_j = (\omega_{0,j}, \omega_{1,j}, ..., \omega_{n,j})^T \tag{7.2}$$

In order to reduce the dimensionality of the original document vector, we utilize the Classical Multi-Dimensional Scaling technique to project document vectors onto a two-dimensional subspace [Dav92]. Given n items in a p-dimensional space and an $n \times n$ matrix of proximity measures among the items, multidimensional scaling (MDS) produces a k-dimensional representation of p items such that the distances among the points in the new space are preserved and reflect the proximities in the data [Fod02]. In our data sample, the input data of MDS is a square matrix containing dissimilarities between pairs of document vectors. The output data is a lower-rank coordinate matrix whose configuration minimizes a loss function called stress:

$$\underset{d_1,...,d_l}{\operatorname{argmin}} \sum_{i<j} (\|d_i - d_j\| - \delta_{i,j})^2 \tag{7.3}$$

where $(d_1, ..., d_l)$ is a list of document vectors in lower dimensional space. $\|d_i - d_j\|$ is the Euclidean distance between documents d_i and d_j. $\delta_{i,j}$ is the dissimilarity value, i.e Euclidean distance, between documents i and j in their original dimensional space.

Given a list of document vectors, using MDS will project the high-dimensional vector on a two-dimensional map such that documents that are perceived to be very similar are placed closed to each other on the map, and documents that are perceived to be very different are placed far away from each other.

7.4.3 SIMILARITY MEASURE

The similarity coefficients between every two document vectors in a reduced dimensional space can be defined as the Euclidean distance between them. Once we have obtained a similarity value for every pair of translations of the same segment, then a weight value for each translation can be computed by averaging the sum of similarity values between the given translation and all other neighbouring translations. As introduced in Section 7.2, we name this value as "Eddy", which can be defined as:

$$Eddy_(D^i_j) = \frac{\sum_{k=1}^{n} \|D^i_j - D^i_k\|}{n} \tag{7.4}$$

where n is the number of documents in a segment i. D^i_j represents a document j in a segment i.

In a traditional clustering algorithm, a diameter refers to the average pairwise distance between every two elements within a cluster [XW05]. If translations of the same segment are regarded as a cluster, then the stability of the segment from the original play can be measured by its diameter. A segment with low stability indicates that translations for this segment vary a lot between different authors, whereas a segment with high stability indicates translations for

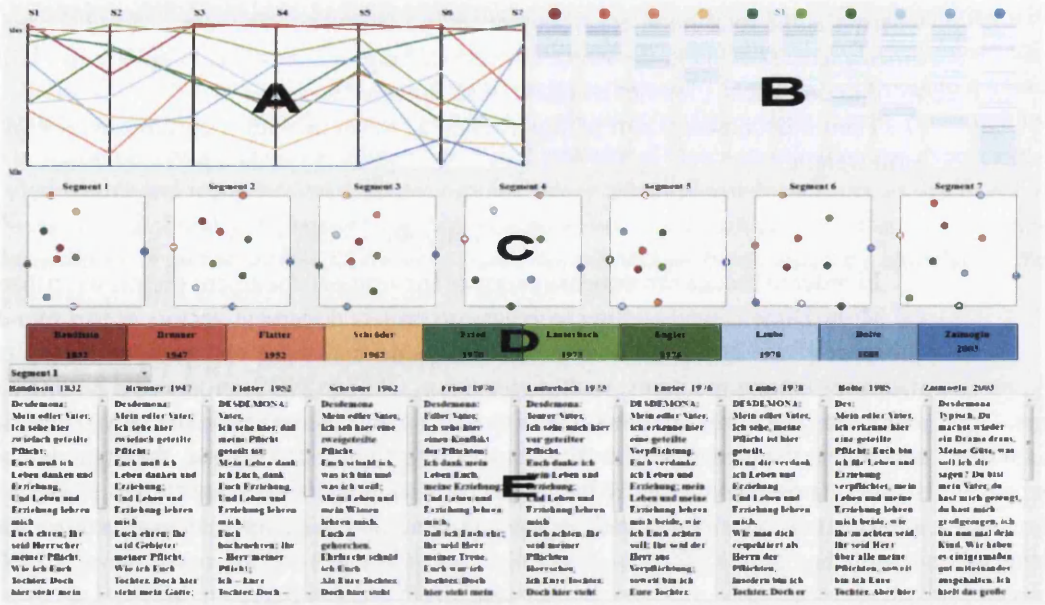


Figure 7.3: This figure shows an overview of our visualization system. (A) is a parallel coordinates view which shows the similarity values for each translation across multiple segments. (B) is the heat map representing the term-document frequency matrix. (C) is a scatterplot view which depicts the relationship between translations in each segment. (D) shows the document control panel where the user is able to brush and select one or many translations for comparison. (E) depicts the actual text.

this segment are similar. As introduced in Section 7.2, we name the diameter for a segment i as "Viv" value:

$$Viv(i) = \frac{\sum_{k=1}^{n} Eddy_(D_k^i)}{n} \tag{7.5}$$

where n is the total number of translations in a segment i. This "Viv" value can be used to rank the segments with respect to the degree of variance between its translations.

7.5 VISUALIZATION

In this section, we present our interactive visualization system to explore and analyze the extracted segment features from Section 7.4. Ben Shneiderman [Shn96] proposed the visual information seeking mantra: overview first, zoom and filter and details on demand, as visual design guidelines for interactive information visualization. Following this rule, our visualization system is composed of two parts. One offers a context view which is composed of scatterplots and parallel coordinates views, which gives an overview of distributions and relationships between translations across different segments, as discussed in Section 7.5.3 and

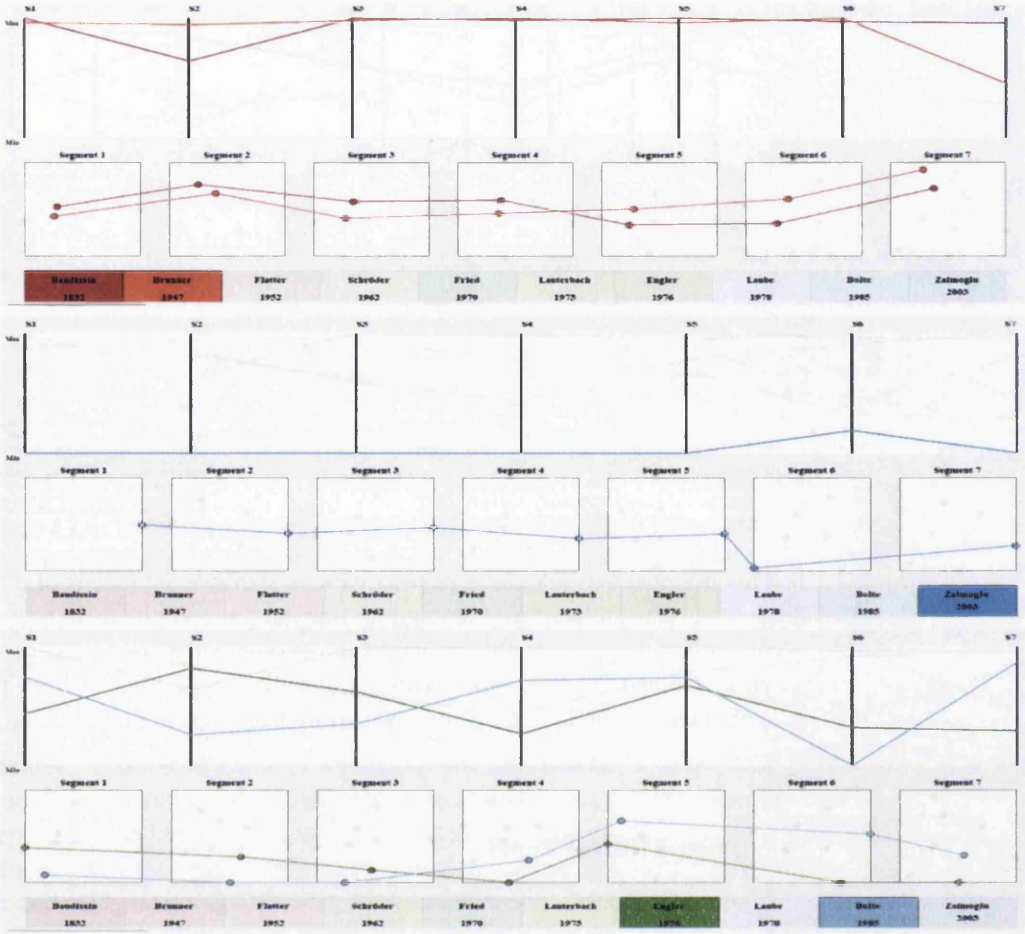


Figure 7.4: This figure depicts three interesting findings by the means of brushing and selection.

Section 7.5.2. The other part provides a detail view, which allows an in-depth analysis for one individual segment using term-document frequency heatmap. This view provides a side-by-side textual and term-document frequency comparison to uncover the underlying details which result in clusters or outliers, as discussed in Section 7.5.4. Shown in Figure 7.3 is an overview of our visualization system. The input data set is a document corpus with ten translations by different authors in different time periods. The details of these translations are introduced in Section 7.3. Each translation can be decomposed into seven different segments. Each segment is an individual speech translated from the original *Othello* play. Different versions of translations have different interpretations for each speech of the *Othello* play, we have therefore built a separate concordance for each segment.

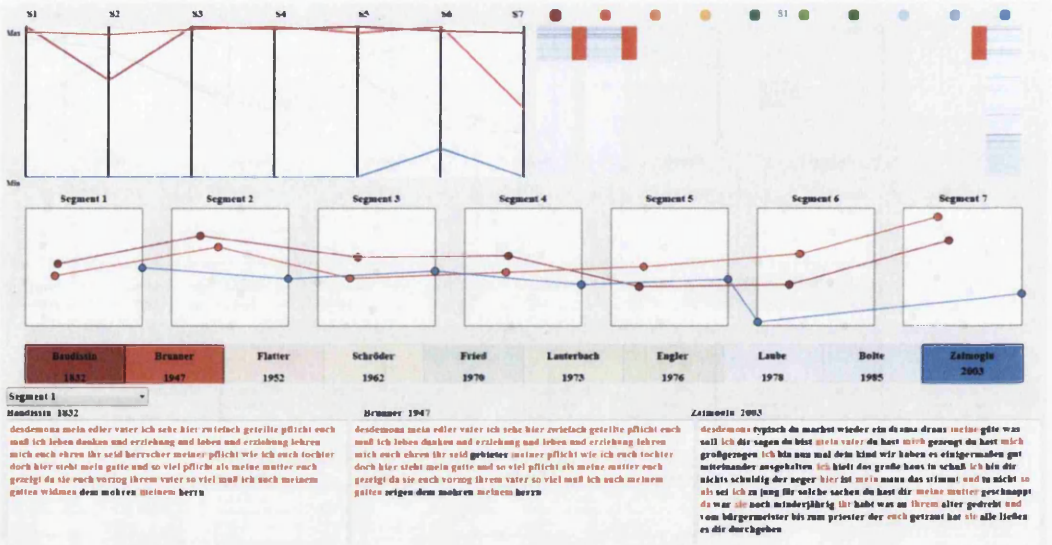


Figure 7.5: This figure shows a focus + context view of multiple selections of different translations. These selections include two very similar translations and one extra translation which appeared as an outlier. The user is able to obtain an overview of segment distinctiveness from the context view. Comparing the corresponding translations side by side from the text view enables in-depth analysis. Unique terms brushed from heat maps are highlighted in red in the text views.

7.5.1 Document Control Panel

Part (D) of Figure 7.3 shows a document control panel. Each rectangular box is assigned a unique color to depict a unique translation. Labelled on the box is the name of the author and the year the corresponding translation was published. The translations are arranged in chronological order by default. The user is able to select one or many translations for comparison. Every time they select a translation, the scatterplots and parallel coordinate views are updated. Interactions on the scatterplots and parallel coordinates make the brushed documents highlighted in the document control panel.

7.5.2 Parallel Coordinates View

Part (A) of Figure 7.3 shows parallel coordinates [ID90b]. Parallel coordinates, introduced by Inselberg and Dimsdale [Ins09, ID90b] is a widely used visualization technique for exploring large, multidimensional data sets. It is powerful in revealing a wide range of data characteristics such as different data distributions and functional dependencies [Kei02]. As discussed in Section 7.4.3, for each translation, an Eddy value is computed for each of its segment. This information can be depicted by parallel coordinates, where each dimension represents an individual segment with every Eddy value linearly interpolated on it. Then an Eddy value for a translation containing various segments can be depicted by a polyline in the parallel coor-

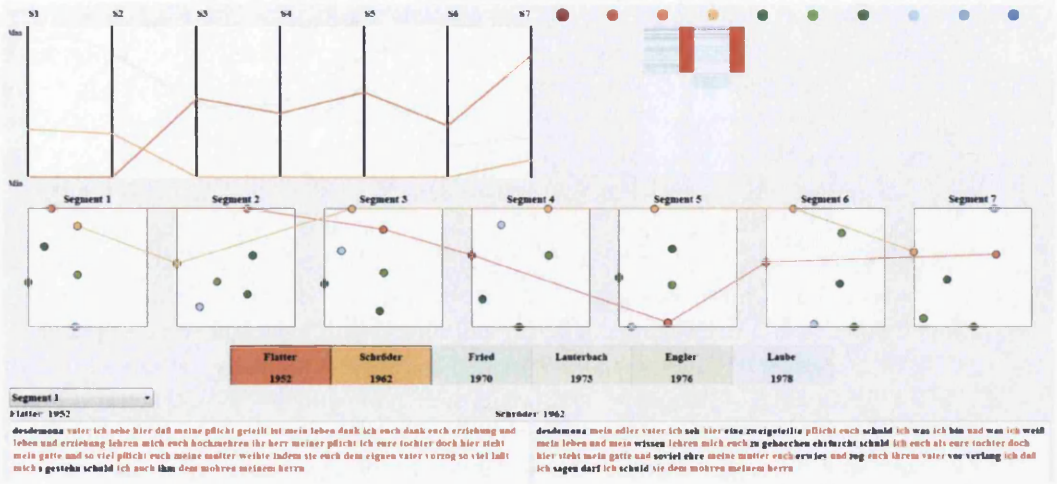


Figure 7.6: In this image, the domain experts have pushed aside some of the uninteresting documents and the rest of the documents are rescaled on the scatterplot and parallel coordinates. Based on this smaller subset and rescaled visualization, the domain experts find two interesting documents, as highlighted and linked in the scatterplot view. These two documents are distinct from the others, especially Schröder appears as an outlier.

dinates. The top of the axis represents the smallest Eddy value, which means on average a translation is similar to all the other translations in a given segment. The bottom of the axis represents the largest Eddy value, which means on average a translation is different to all the others. We offer various interaction support, such as an AND and OR brush, for the user to explore different multidimensional patterns.

7.5.3 Scatterplot View

The parallel coordinates view presents an average similarity value for each translation across multiple segments. If the user is interested in the relationship between each pair of translations for a given segment, we incorporate multiple scatterplot views to represent this information. Document vectors with reduced dimensionality can be visualized and presented by scatterplots for each segment, as shown on part C of Figure 7.3. Each translation is depicted by a constant unique color across all segments. The scatterplots offer a clear overview of how different translations relate to each other. The relative positions of document vectors in the scatterplot can visually reveal which set of translations are close to each other and which are further away. This could additionally uncover some interesting clusters or outliers. For example, we are able to observe an outlier as depicted in blue on the far right of segment one and on the top of segment three. In addition, from the parallel coordinates view, we are able to see that this translation written by Zaimoglu in 2003 is an outlier across most of the segments, which draws the same conclusion as our initial assumption. For some of the segments, documents are almost equally distributed and not positioned closely as a compact cluster, such as segment

six and seven. These segments have a relatively larger pairwise Euclidean distance between translations compared to other segments. This indicates that authors might have distinctive interpretations for these two segments in *Othello*. If the users would like to see how a whole translation behaves across all segments, then we provide a link to connect the corresponding point in each segment scatterplots, as shown on the top of Figure 7.4. This provides a coherent view of how similar each translation is compared to others in each of its segments. Figure 7.4 depicts several interesting initial findings by the means of brushing and selecting as discovered by domain experts. The first finding is shown in the first row of Figure 7.4, which shows the closest similarity between Baudissin and Brunner: editions of the same text, with orthographic differences in all segments and term- and phrase-differences in some segments. The second finding is shown in the second row of Figure 7.4, which clearly identifies the stylistic outlier, Zaimoglu 2003, a very idiosyncratic translation or 'tradaptation'. The third finding is shown in the third row of Figure 7.4, which demonstrates that the two didactic prose translations for study purposes (Engler 1976, Bolte 1985) cluster together in most segments, distinct from all others. This is expected: these versions share the same time period, translation skopos (purpose: didactic), and aesthetic form (prose), all leading to similar word-choices. As the translations are selected, the corresponding document is shown to give a side-by-side textual comparison. As illustrated on the part (E) of Figure 7.3. Once the user has observed some interesting patterns from the context views, they can zoom into each segment for more detail from this text view.

7.5.4 Term-Document Frequency Heat Map

The system created here was done in close collaboration with a domain expert in German translations of Shakespeare's work. The following review is provided by him. When we checked varying distances on the scatterplots against actual textual differences, we discovered that significant differences in word-choices are not easily identified. Distances are computed from concordances which treat different word-forms as different tokens (e.g. 'Cypern'/'Zypern', 'kräftigen'/'kräft'gen'). Therefore only relying on the scatterplot and parallel coordinates view is not yet effective for identifying segments where translators (and editors) of very closely similar versions make different significant word-choices. In order to analyze differences between pairs of versions in more detail, including a measurement of character-string similarities (which also will help detect genetic relations), we have proposed a term-document frequency heat map to compare segments on term level. Part B of Figure 7.3 is a term-document frequency heat map for segment one. Each column of our heat map represents an individual document. For a better discrimination between different documents we decide to leave a small gap between every two columns. Each row of our heat map represents a unique keyword. Every cell inside a heat map depicts the frequency of a keyword (row) in a given document (column). The darker color in each cell reveals a higher term frequency and the lighter color reveals a lower term frequency. Our keyword list contains all the unique words occurred in all translations in this given segment. From this heat map, we are able to easily observe that the first two segments share a number of common words. This might explain why these two segments stay closer to each other from the scatterplot view described in Section 7.5.3. In addition, the user is able to brush these common keywords and the corresponding document text view will be updated,

as shown in Figure 7.5. The text view shown on the bottom row of figure 7.5 depicts three selected document in segment one. The brushed keywords from the heat map are highlighted in red in the text view. As we can observe that the first two translations are very similar with respect to the common words and sentences they share. However the other selected documents only share a few of the brushed keywords and reveal a different style of writing. A full list of heat maps for all of the segments is shown in Figure 7.7.

7.6 DOMAIN EXPERT REVIEW

The ShakerVis tool implements a new approach in textual studies: comparison of multiple translations, which have been segmented and aligned, using metrics to analyse the relations among lexical choices in translations of individual segments. The point of doing this is that multiple translations of great works of world literature, philosophy, and religion are rich data sources for arts and humanities research, but so far under-exploited. The scriptures of all major religions, influential ancient and modern philosophical works, and important works of literature are in many cases translated over and over again into major world languages, each time differently. Such re-translations all embody variant interpretations of their source texts. They document cross-cultural relations between source and target cultures, and they document the evolution of language and ideas in target cultures. That makes them very significant sources. But even beyond this, the patterns of variation among translations can also shed new light on translated texts themselves. Literary, religious and philosophical texts are essentially polysemic or ambiguous: they can be interpreted in various ways. By studying the various ways in which they have been interpreted by translators , we can discover important aspects of their meaning-potential, which would not be obvious if we only read them in one language, or only read a few of the many existing translations. Thus, both diachronic (historically-oriented) and synchronic (trans-historical, comparative) approaches to multiple translations are appropriate. ShakerVis enables us to advance investigations of both sorts.

Until now, in print media, comparing large numbers of translations in systematic ways was a very difficult and tedious task, which took huge amounts of scholars time, and the findings could not be easily presented or verified. As a result, studies of multiple translations are few and far between, and the researchers tend to select only modest numbers of translations, and to present only small selected samples to the readers of their research publications [Seh09]. Our work is seizing the opportunities presented by digital media to create new tools which facilitate comparison of arbitrarily large sets of translations, in their entirety, and collaborative investigations of them by teams combining different disciplinary and linguistic skills. We aim to make the processes of creating versions corpora and exploring variation within them far easier, and to facilitate the formulation and investigation of hypotheses and the presentation of findings. Some prototype tools are presented online at www.delightedbeauty.org/vvv [CFT12]. We intend to integrate the key features of ShakerVis with our online work.

ShakerVis is an important prototype for further development of our approach. It allows us to explore patterns in variation among multiple translations (versions) of a text, from segment to segment. The color codes associated with individual versions provide clear visual navigation between versions and the visualizations of their inter-relations: scatterplots and paral-

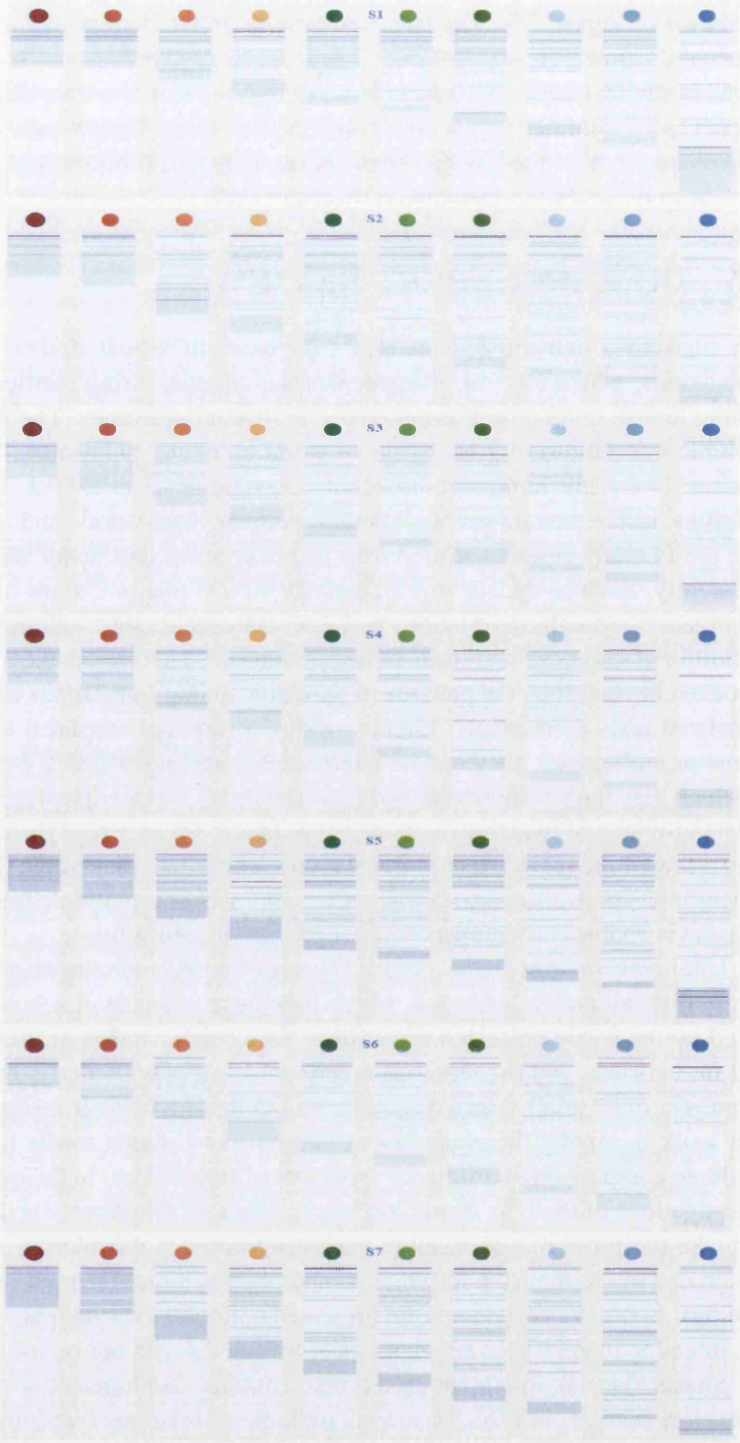


Figure 7.7: This image shows the term-document frequency heat maps for all of the seven segments.

lel coordinates, offering alternative representations of relations of proximity/distance between word-choices per segment. The scatterplot view of differences is more useful than the parallel coordinates view. Full text view is important so we can check analytically discovered patterns by reading actual text data. A limitation of the interface, dictated by desktop screen size, is that only 10 versions can be compared. Our current dataset includes 37 German versions of Shakespeares Othello, and even that is only about half the extant German translations/adaptations. The ShakerVis experiment only tackled 7 segments (speeches) in the play: our dataset includes over 80, and even that is only about 10% of the play. As our work develops, the problems of scale, which obstruct translation comparison in print media, also become more problematic in digital media. We eventually hope to work with translations in as many different languages as possible: in the case of a popular Shakespeare play like Othello, that would mean around 400 translations in 100 languages. (No reliable global census of Shakespeare translations even exists.)

As discussed in Section 5 above, Figures 4, 5 and 6 depict several interesting initial ndings by the means of brushing and selecting scatterplots and parallel coordinates in ShakerVis. A first set of findings confirms what we already know about the texts, and this reassures us that the patterns being discovered by the tool and the underlying metrics correspond with ground truth. Two translations [Bau32, Bru47] are variants of Baudissins famous 19th-century translation: they are absolutely similar in wording, except for orthographic differences and some changes in wording made by Brunner as editor. Two translations [Eng76, Bol85] are both generically and historically similar to one another, and distinct from all the others, in that they are didactic prose translations of the 1970s-80s, for classroom use. (The other eight are translations for stage performance and/or for general readers.) As we would expect, ShakerVis shows each of these two pairs of versions clustering, in all segments, more than any others. Where Baudissin and Brunner are concerned, ShakerVis scatterplots also show different distances from segment to segment, depending on what proportion of words in the segment differ (Brunners different word choices or different orthography). Finally, another expected finding is that the most free translation of all, Zaimoglus controversial recent tradaptation using modern slang, shows up in ShakerVis as an outlier in all segments. Zaimoglu [Zai03] uses different wording from any other translation. These results are not surprising, but welcome confirmation that the tool is in principle reliable.

Further partial confirmation is provided by the result depicted in Figure 6. Previous non-digital, but quantitative-algorithmic work on over 30 German translations of a single segment in Othello (the rhyming couplet: If virtue no delighted beauty lack, Your son-in-law is far more fair than black) identied Schröders translation as the most distinctive of all (i.e. the highest Eddy value). The modified algorithm used in our online Translation Array places Schröders translation of this segment as the second most distinctive [CFT12]. In ShakerVis, when we re-scale the sample of ten versions analysed to exclude the five just mentioned (the two variants of a 19th-century translation, the two didactic translations, and the 21st-century outlier), we are left with versions of the 1950s-1970s, all written to be performed, and in verse: Flatter, Schröder, Fried, Lauterbach, and Laube. These are historically and generically similar, but diverse in their wordings. Among these, ShakerVis scatterplots and also the parallel cordinates show Schröder as a clear outlier in most segments (i.e. highest Eddy value), followed by Flatter as the next most distinctive. So Schröders relative distinctiveness as a translator, found in some

previous work, is confirmed in this different sample. However, it must be added that Schröder does not appear as a particularly distinctive translator when all Eddy values for all segments in our online dataset are averaged (Eddy History graphic, in [CFT12]). Of course this underlines the importance of a systematic and wide-ranging comparative study, and the limitations of sampling, where literary texts are concerned. The ShakerVis analysis must be extended to our full text existing dataset, and indeed other, larger datasets.

ShakerVis also produces more surprising discoveries, which raise new research questions: exactly what we aim to do. A first set of questions relate to translation genetics (translations depending on or borrowing from earlier ones) and translation periodisation (translations obeying cultural rules of style specific to certain historical periods). Setting aside variant texts, which are known to be close genetic relatives, and a few versions which are explicitly identified as being based on an earlier translation, most translations are presented as the translators original work; but in fact in most cases the translators knew, and probably re-used, the work of previous translators. Just how they did so is interesting to humanities researchers from several points of view. An interesting ShakerVis result is the finding that the translation by Fried (1970) [Fri99] appears closest (of all others in this sample) to the two didactic prose versions, clustering with them in most segment scatterplots. The didactic versions (1976 and 1985) are later than Fried. A periodisation effect a certain style of translation from the 1970s and 1980s can be excluded here, because other translations in the ShakerVis sample, from the same decades, do not show the same proximity. Periodisation effects could be systematically investigated with a larger sample: we know that such effects exist, but we do not know exactly how they work. It is more likely in this case that the didactic versions were directly influenced by (i.e. borrowed some wording from) Frieds version. The concordance heatmaps do not particularly help us to investigate this hypothesis, as they display all words used by all versions, and do not highlight multiple specific words which are re-used by multiple versions, nor do they allow us to select multiple non-neighboring words. Signals of significant word re-use which would be expected in cases of borrowing therefore remain hard to detect amid the noise of variation. There is room for refinement here. But, alerted by scatterplot proximity, we can read and compare the versions, and we can then see that the didactic versions by Engel and Bolte do, indeed, have some wording in common with Fried which is not found in other versions. We still have some way to go in this area, but hypotheses concerning genetic relations can be investigated far more efficiently and tested far more accurately with digital tools than by means of arduous close comparative reading alone.

Frieds version is involved in two more findings. ShakerVis scatterplots show a tendency for Fried to cluster with other post-1970 versions (as well as the didactic versions), in some segments. If this can be conrmed as a trend with a larger data sample, it raises interesting questions. Frieds translations of Shakespeares plays were very prestigious in German culture in the 1970s-80s, and are still highly regarded, in print and used in theatres, today. But they were and are not the only prestigious Shakespeare translations, by any means, over these decades. Prestige can be measured in many ways, but not least in terms of influence on other translations. If we can determine patterns in borrowing between translations, we can create an algorithmically-generated time-map of translation genetics, influence and relative power: a map which shows how different translators work relates to that of their precursors and successors. This would be an important contribution to understanding the evolution of the culture concerned. To do this,

we might want to filter out periodization effects, in order to isolate clusterings only explicable in terms of textual genesis. This kind of analysis and output would be interesting in many other re-translation contexts, as well as Shakespeare.

In fact, in a culture where there are very many different translations of a particular work, questions of borrowing are highly controversial, because translators intellectual property is involved. Hamburger [Ham06] discusses this question passionately with reference to German Shakespeare translators, particularly mentioning cases of translations used in theatres in the former East Germany in the 1980s, which were based on West German translators work (such as Hamburgers), without permission or payment of royalties. So it is very interesting indeed that ShakerVis scatterplots show the work of East German translator Lauterbach (1973) [Lau96] clustering more than any other stage version in this sample with Fried (1970) [Fri99]. From simply reading the two texts side by side, it would not appear obvious at first that Lauterbach has borrowed from Fried. But after ShakerVis points us to this proximity, we read and compare these versions again. Now, certain similarities are striking. As with the didactic versions, once we have been alerted to it, we can see that Lauterbachs version has some wording in common with Frieds. Whether this might be due at least in part to a periodisation effect, or a genetic effect (i.e. borrowing, even plagiarism), is an interesting topic for further research.

Perhaps the most interesting result of the ShakerVis experiment relates to the question of differences between segments in the translated text, in terms of translators aggregated behaviour: that is, a Viv value finding. Even though the sample is small and the method experimental, ShakerVis appears to have enabled us to discover an Othello Effect in translators aggregate choices when re-translating a great work. ShakerVis allows us to investigate the hypothesis that translations in general (in any one language, at least, and possibly also across multiple languages) vary in regular ways according to specific variable features of the translated segments. This could apply to many kinds of features, including differing levels of difficulty, ambiguity, or obscurity of meaning, or ideological contentiousness. Such features of discourse are hard to define objectively or quantify, not least because they may be considered as intrinsic to a translated source text, or else as properties of the relation between the source text and the translating and interpreting culture. They may, however, become definable through refinements of the analytic approach we are developing: that is a key aspiration in our work. On the other hand, features such as speech by [character name], are simple, objective attributes of segments in a dramatic text. And it is more than likely that translators, as a group, tend to respond differently to different characters, i.e. speakers in a dramatic text, whose speaking parts are each represented by a different set of speech-segments. So speaker attributions are a suitable focus for investigating possible regularities in associations between segments with specific features (in the translated text and all translations), and regularities in the range and distribution of Eddy values calculated for all translations. We refer to the quantification of such ranges and distributions as Viv values [CFT12]. They represent the amount of divergence between all the translations of a segment, or the overall stability/instability of the translations . A segment which most translators translate with similar words has a low Viv value. Where translators seem to disagree with one another a lot, Viv value is high. This is a way of pinpointing segments in a text which provoke dissent among translators: where there is greatest interpretative variation across all the translations. For humanist readers of great works, this is potentially very interesting as a way of detecting hotspots of disagreement over what a text

might be said to mean. It also promises to provide new kinds of evidence of what exactly translators do when they translate differently from one another. In our online prototype work, Viv values for segments are calculated from all Eddy values by various experimental metrics (as an average of the Eddy values, or as their standard deviation) and displayed as a varying color coding, underlying the base text (i.e. the English Shakespeare text) [CFT12]. ShakerVis does not represent Viv values as such, but the scatterplots can be read as indicators of Viv: Viv is highest where the distances are greatest, i.e. there is least clustering. This is visually intuitive and effective. It turns out that ShakerVis provides evidence of an Othello Effect, visible in Figure 3, which is highly interesting for the study of literary translations.

The sample of seven segments from Othello was chosen to include seven speeches by: Othello, the plays hero (segment 6); Desdemona, his wife (1 and 7); Brabantio, her father (2 and 4); and the Duke of Venice (3 and 5). The expectation was that Desdemonas speeches would be more variously translated than others, because the interpretation of her speeches in the sample is known to be controversial: her character, her behaviour, her values as presented in the play are a topic of much debate, and her specific speeches in this sample provoke disagreements among critics and other interpreters (including directors and actors, and presumably translators). In Figure 3, we see the scatterplots for all seven segments and all ten versions. The changing variation and clustering seems random. As for Desdemonas segments, segment 1 shows quite a lot of clustering; segment 7 shows greater distances. But (in this small sample) there is no sign of a Desdemona Effect a collective tendency to translate her speeches more variously. Instead, with all due caution due to the small sample size, it looks as if we may have an Othello Effect. In segment 6, the distances between all versions are greatest: six of ten versions are at the sides of the scatterplot, and four others are almost equally distant from them and from one another. This segment is the only speech in the sample by Othello, the hero of the play. It seems that in this speech, the selected translators have most differentiated their texts from one another, whether consciously or not (most translators knew some other translations, but none of them knew them all). As before, the findings suggested by the tool need to be checked by close reading. Recall that this sample includes two variants of Baudissins famous version: Baudissin and Brunner. On re-reading them, it becomes clear that when Brunner edited Baudissins text, in segment 6 he went to greater lengths to alter Baudissins version than he did in other segments in the sample. The two didactic versions, generally rather similar, are also more different from one another in segment 6 than in other segments. The outlier, Zaimoglu, is less distant from all others in the segment 6 scatterplot than in other scatterplots, not because he translates segment 6 more similarly to any other version, but because the other nine are all more distant from one another in segment 6 than in other segments. When we use the tool to re-scale the sample of versions, while still comparing all segments, e.g. by excluding the Baudissin pair and/or the didactic pair and/or Zaimoglu, the Othello Effect appears to persist: in this segment, the translations are least stable, or have highest aggregate distance from one another highest Viv value. Like all the other results of the ShakerVis experiment so far, the Othello Effect needs to be conrmed by analysing a larger sample of versions and segments, more texts and in more languages. We plan to do this in future research. But ShakerVis has enabled us to establish a new, plausible and investigatable hypothesis: in multiple re-translations of a play text (and perhaps also in re-translations of other speaker-based literary texts, such as dialogue-rich or multi-perspectival fiction, or philosophical symposia), the level of overall

variation in speaker-associated segments relates to the perceived importance of the speaking character. Here, importance may be a quantifiable factor, based on how many words and in a play how many speeches are associated with the speaker. For a more important speaking character, we hypothesise, translators tend to make more investment of thought and imagination to remake the words in their own way, compared to rival translators. This hypothesis is in accord with studies of re-translation based in Bourdieus concepts of distinction and cultural capital, which depict re-translators as being in a state of implicit struggle with one another for social and cultural standing [Han05]. But such studies tend to draw evidence chiefly from paratexts (translators self-justifying introductions and comments). It is new and exciting to find that digital tools make it possible to explore translators implicit struggles with one another, using the evidence of the actual fabric of their translations.

ShakerVis, particularly when we have integrated its key features with our online tools, will make important contributions to increasing knowledge and developing new theory in the innovative area of visualization-based re-translation corpus study, which has the potential to open important new horizons in the exploration and analysis of major works of world culture.

Chapter 8

Conclusion and Future Work

Contents

The objective of this thesis is to address one of the intrinsic challenges in data visualization when handling large and high-dimensional data sets. In order to achieve this goal, I have proposed several novel techniques and extension to parallel coordinates. Throughout this thesis, I work closely with researchers from other domains and have applied the proposed visualization techniques to their collected data sets. Chapters 3, 4 are mainly focused on visualizations for large data sets, in which an in-depth visual analysis is performed on animal tracking data sets. Chapters 5, 6 and 7 are mainly focused on high-dimensional data sets. These data sets come from the derived metadata of German translations of *Othello*. My proposed techniques and visualization systems are evaluated by the domain experts who are studying the particular problem. To conclude this thesis I again re-emphasize the main benefits and contributions in each chapter:

- In Chapter 2, I have surveyed a range of off-the-shelf, freely available information visualization tools for the visual analysis and investigation of document triage. Although there are many options available, only a select few visualizations are useful for this particular application. The beneficial visualizations are able to get more insight of the data and make the new hypothesis and findings specified in Section 2.2. This survey also serves as a useful tool for readers interested in gaining an overview of existing, free, state-of-the-art information visualization tools. I also report positive and negative feedback from experts in the HCI and digital library domain. Using these beneficial visualizations they are able to see new properties of their document triage data and formulate new hypothesis which may be helpful for the future experiment design.

- In Chapter 3, I have proposed the angular histogram and attribute curves for visual analysis of large and high dimensional data. My method is based on the vector-based binning which not only depicts the data distribution but also reveals the angular information of the polyline-axis intersections. Therefore the angular histograms and attribute curves of-

fer an information-rich overview. Also I provide various interactions for the user to select and brush the interesting subset of the data sets. I compared and evaluated my methods with the line-based histograms [NH06] and alpha-blending with respect to cluster analysis, linear correlation detection and outlier analysis. I demonstrated my technique using real world animal tracking data set.

- In Chapter 4, I have developed a Markov Chain model for visualizing multidimensional patterns with parallel coordinates. A histogram or scatterplot view presents the joint probability distributions for all data samples. The user is able to brush the data trend based on this probability value. Using my method, the global data trends in higher dimensional space can be discovered and displayed. When rendering a large data set, I propose an extended angular histogram to represent the data density. I also demonstrate a case study on marine biology data.

- In Chapter 5, I have surveyed and compared a range of off-the-shelf, freely available information visualization tools for the visual analysis and investigation of the collected *Othello* data set. Although there are many options available, only a select few visualizations are useful for this particular application. This study also serves as a useful tool for readers interested in gaining an overview of existing, free, state-of-the-art text visualization tools for text analysis.

- In Chapter 6, I have developed an interactive visualization system for exploring the variation on term level among different German translations of Shakespeare's play, *Othello*. A structure-aware treemap is developed for metadata analysis and the focus + context parallel coordinates is developed to investigate the variations among the translations. Parallel coordinates incorporate an objective similarity measure for each document using LSI model. Also, various interaction supports are realized to facilitate the information seeking mantra: overview first, zoom and filter and detail on demand. My visualization is evaluated by the domain experts from Arts and Humanities.

- In Chapter 7, I have derived statistical metrics, such as Eddy and Viv value to measure the stability of segment translation of *Othello*. Based on these metrics, I'm able to develop an interactive visualization system for analyzing segment variations between German translations of *Othello*. My system is composed of two parts, one is the context views which utilize parallel coordinates and scatterplots to explore variations between multiple segments. The other part is the detailed views including the term-document frequency heat map and textual visualization to compare different translations in the same segment. My visualization system is evaluated by the domain experts and help them explore some interesting findings.

8.1 Future Work

There are still a number of issues remains as challenges in parallel coordinates. Finding the optimal axis ordering is one of them. Changing the ordering of the parallel axis often leads to different multidimensional visual patterns. Although the intrinsic underlying data is not

changed, visual perception with respect to knowledge discovery process can be affected by axis re-ordering. An open issue then arise is whether an optimal axis ordering is existed which offers the user the best visual perception to identify the important data features. If such an order is existed, can we find it? This is a very challenging problem especially when the dimension of the data becomes high. In chapter 3, we have proposed a Markov Chain model to present the multidimensional patterns. Because in our model, the probability distribution is optimized for different orderings. In the future, we plan to extend our model with automatic axis reordering of parallel coordinates and use an adaptive binning according to a pre-defined criteria. The other issue of parallel coordinates is the scalability both to dimensionality and quantity of data samples. Too high a dimensionality or too large a data quantity will quickly exhaust the screen space. There are a large number of previous work on dimension reduction. However, from the perspective of visualization, the question is can we find out a better visual representation for conveying very high-dimensional data ?

As for the *Othello* data set, future work includes analyzing a larger corpus of 88 (or more) segments and 32 (or more) versions. This will add challenges for user navigation. We also need to work with non-contiguous, nested and overlapping segments and one-to-many segment alignments. We must combine the selecting/filtering options in this visualization with those offered by other Translation Arrays interfaces (e.g. segments grouped by speaker, length).

Appendix A

Appendix: Force-Directed Parallel Coordinates

Parallel coordinates [ID90a] remain a fundamental technique for information visualization. In particular, parallel coordinate plots (PCPs) allow multidimensional data sets to be presented in a natural manner without loss of information, and have been gainfully applied to many such problems. The problems of occlusion when displaying large data sets and of determining an appropriate axis order to highlight multivariate relations have been the subject of much recent research. In applying parallel coordinates to real world problems, though, Inselberg stresses that interactivity is essential [Ins08].

In exploratory analysis, the role of visualization is often not to answer specific questions, but in determining what questions might have interesting or useful answers. Direct and indirect manipulation techniques are vital in this process, but the set of such interaction techniques applicable to PCPs is currently limited. Axes can be brushed for selection or filtering, as can the spaces between axes, and the order of axes can be changed, algorithmically or through user interaction. While these are powerful, a richer set of interactions could improve the exploration process, and provide the user with a better understanding of the underlying data.

Another motivation behind our work lies with interpretation. New users often have difficulty interpreting parallel coordinate visualizations and understanding their benefits when seeing them for the first time. The model we present helps facilitate interpretation. One obvious visual comparison for a PCP is with a bridge. This is a useful analogy for our work because a cable-stayed bridge consists of cables connected to towers to support a road. This is analogous to a PCP where the axes of the plot correspond to the towers of the bridge, and the cables to the projected data points. Bridges, though, are much more familiar objects: the stresses and tensions in the towers and cables that compose these structures are well-understood.

In this appendix, we present force-directed parallel coordinates, a technique that uses a familiar visual metaphor to represent data in PCPs by modelling the plot as a physical system, together with some physical interactions with the user. We contribute:

- A novel visual metaphor for parallel coordinate plots as a physical system to aid intuitive understanding of the underlying data

- Three new interaction techniques that act in concert with this metaphor to aid with exploration and the occlusion problem

We demonstrate the benefit of this novel model and interaction on a number of both synthetic and real world data sets, and conduct a user study to evaluate the technique.

The remainder of this paper is organised as follows: work from a number of related fields is reviewed in Section A.1. In Section A.2 we describe the mathematical model for the physical system formed from the plots. Novel interactions with the data through this system are described in Section A.3 and demonstrated in Section A.4. The methodology and results of a user study on usability is presented in Section A.5 and these results are discussed together with suggestions for future work Section A.6.

A.1 Related Work

While there is a vast amount of literature on parallel coordinates in general [Ins08], we concentrate on work that focuses on interaction with PCPs. Interaction techniques are summarised in work by Kosara et al. [KHG03a] for information visualization general and Siirtola [SR06] for parallel coordinates in particular. Here, we consider recent research in two areas specific areas of PCP interaction — brushing and axis re-ordering — before moving on to examine other uses of force-directed techniques in information visualization.

Brushing, the act of selecting of a set of polylines within a PCP on which to perform an operation such as zooming, deleting, highlighting or masking, has been referred to as the fundamental interaction with parallel coordinate plots [SR06]. It can occur on or between axes - Inselberg [Ins08] refers to these as *Interval* and *Pinch* respectively. By combining brushing on multiple axes with logical operators brushes can act in multidimensional fashion [MW95], as in the XmdvTool system [War94b]. Polyline segments can also be selected based on their correlation, since in PCPs correlation is depicted as the slope between axes [HLD02a]. This allows a more nuanced selection than selection on a single axis and facilitates identification of outliers.

Allowing selection directly on the plot, rather than providing a separate brush interface, can enhance the process by providing continuous feedback [Sii00], and modern hardware may enable a more natural interface for this direct manipulation [Kos11]. Brushing can be used to provide increase display resolution for only the brushed region [WB96b] and also has been shown to be feasible for large data sets [BBP08a]

Axis re-ordering is an important element of exploring data sets using parallel coordinates, since correlations or relationships are only visible between adjacent axes. While for a parallel coordinate plot of n dimensional data there are $n!$ possible axis permutations, only $\lfloor (n+1)/2 \rfloor$ are needed to ensure adjacency of every pair of axes [Weg90b]. However, multidimensional correlations are still easily overlooked and hence determination of useful and appropriate axis orderings is an active research field.

Typically, axis ordering algorithms operate by defining a metric for desirability of layout, and then using an algorithm to find axes orderings that optimise this metric. Dimension arrangement has been shown to be an NP-complete problem [ABK98], so heuristic algorithms such as the ant system algorithm [ABK98], random swapping, nearest neighbor,

greedy [PWR04] or branch-and-bound [DK10a] are used. Similarity [ABK98, TAE+09], clutter [PWR04] and screen space [DK10a, JC08] have all been used as metrics for these algorithms, which may produce a range of possible useful orderings.

Force-directed algorithms have been widely used in graph layout [FR91, DBETT94] to produce aesthetically pleasing representations by performing simplified simulations of physical systems. A system called RadViz [HGM+97] uses Hooke's Law to layout points based on spring forces. Hierarchical edge bundling [Hol06] uses a bundling strength parameter as a measure of stiffness, while Tominski et al. [TAvHS06]'s Bring Neighbor Lens uses an attraction approach to show neighbors to a given node or set of nodes.

The force-directed approach has some significant challenges: high computational complexity ($O(n^3)$)and lack of predictability of layout for similar graphs [HMM00]. The algorithm may never reach a stable state or may take a long time to do so. There may also be several stable states for the system, and local minima may prevent it reaching the most globally stable one. We address this challenge primarily with interaction in our work.

A.2 Physical Modelling

In this section, we describe the physical model by considering line segments as springs and axes as solid rods fixed on a pivot, together with a Verlet integration scheme.

A.2.1 Spring Forces

Consider one spring, with end points $\vec{a}$ and $\vec{b}$, spring constant k and rest length l. The length of the spring, d, is then $|\vec{b}-\vec{a}|$. The force exerted by the spring is, by Hooke's Law:

$$F = -k(\frac{d-l}{d}) \tag{A.1}$$

Since the spring is attached at both ends, $F/2$ acts on each endpoint, and hence on each axis-rod. The rest length for each spring is set to match the inter-axis spacing.

A.2.2 Axis forces

As the rod is fixed at a pivot point, $\vec{p}$, the turning force (torque, τ) on the rod from each spring is dependent on both the distance from the fulcrum and the angles of both the rod and the spring, since only the component of the spring force acting perpendicular to the rod has an effect. This component, r, is given by:

$$\vec{r} = v - \frac{\vec{v}\cdot\vec{u}}{\|\vec{u}\|^2}\vec{u} \tag{A.2}$$

where $\vec{v} = \vec{a} - \vec{p}$ is the vector from p to a and $\vec{u} = F(\vec{b}-\vec{a})$ is the force at point a.
The torque τ on the axis-rod from this component is then simply

$$\tau = \|\vec{r}\|\|\vec{v}\| \tag{A.3}$$

The resultant torque is calculated by summing over all springs on each side of the pivot. Figure A.1 shows an example: spring 1 has $d = r$ and hence exerts no force via A.1. Spring 2 exerts a force on both axes, while spring 3 exerts force only on the first axis, since for the second axis $\| \vec{r} \| = 0$

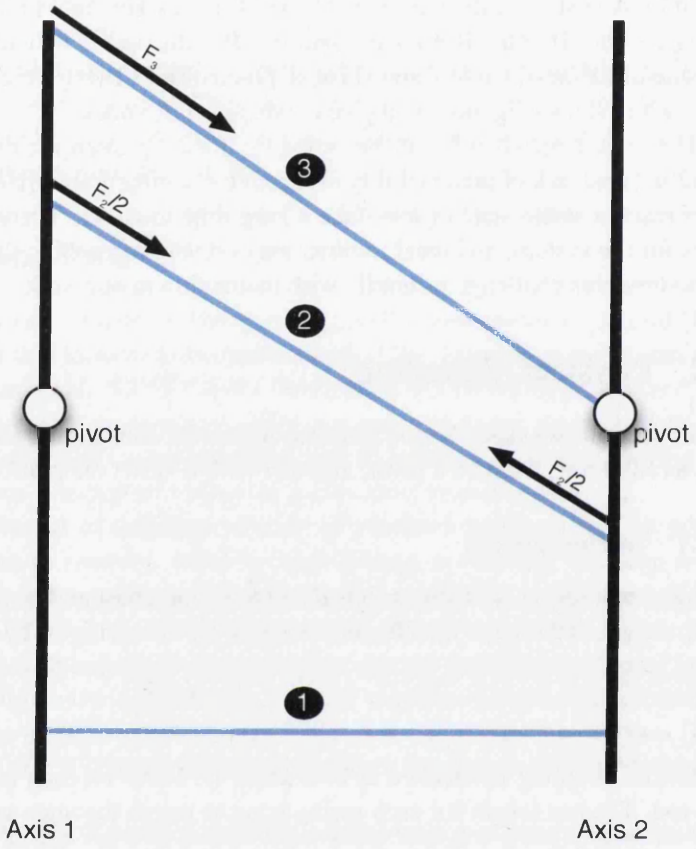


Figure A.1: By considering the axes as rods, the lines between as springs and fixing a pivot point, we can determine the forces on each axis from each spring and hence the resultant angular acceleration of the axes. In this diagram, spring 1 is at its rest length and hence exerts no force on either axis. Spring 2 exerts a force on each axis proportional to the length of the spring and the distance from the pivot. Accordingly, spring 3 exerts a force on axis 1 but no force on axis 2, since the distance from the pivot is 0.

A.2.3 Numerical Integration

Since we are concerned more with stability and convergence than accuracy, a Verlet scheme [Ver67] is used to perform the integration. We determine angular acceleration α via the rotational form

of Newton's Second Law:

$$\tau = I\alpha \tag{A.4}$$

where I is the moment of inertia, and then apply:

$$x' = 2x - x^* + \alpha \cdot \Delta t^2 \tag{A.5}$$

$$x^* = x \tag{A.6}$$

where x^*, x and x' are the previous, current and new positions of the end point of the rod and Δt is the time step.

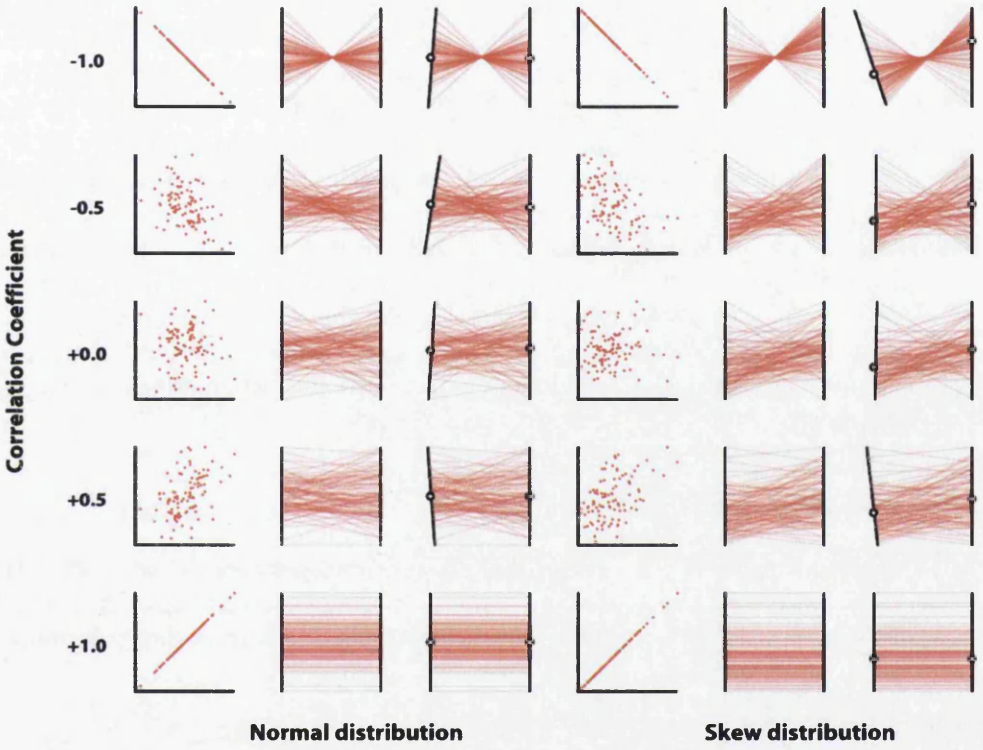


Figure A.2: Comparisons between scatterplots, PCPs and force-directed PCPs for two dimensions using synthetic data generated with known skew and correlation. The first three columns show data with the given correlation where both variables follow a normal distribution, while the last three columns show data for which one axis has a skew-normal distribution and the other is normally distributed. In neither case is there a large movement of the axes, due to our choice of pivot position, and this places the focus for exploration on user interaction.

A.2.4 Pivot Position

The pivot for each axis can be placed at an arbitrary point on the length of the rod. Placing it at either extreme (top or bottom) means the axis rotation is in an inward direction with respect to the next axis. Placing the pivot in the center of the rod gives a system most akin to a cable-stayed bridge. However, placing the pivot at the mean value for that axis gives some interesting results: for a data value mapped to a location x on the rod, the distance to pivot is then $x - \bar{x}$. This is equivalent to mean-centering the data. For this paper, we typically set the pivot for each axis at the mean but also allow the user to choose to use the mid-point of the axis instead, to match more closely with our bridge metaphor. We demonstrate plots using both pivot positions in Section A.6.

Axis rotation occurs only when the resultant torque (clockwise vs counter-clockwise) is non-zero. By mean-centering the data, the rotation that occurs in the system without user interaction can be minimized. Figure A.2 shows some examples of correlated data displayed in three different forms: scatterplots, PCPs and force-directed PCPs. Even for skewed, highly correlated data, little axis rotation is observed. Since rotation is directly observable by the user, this reinforces the physical nature of the system while avoiding distracting the user (and complicating interactions) by having continual axis rotation.

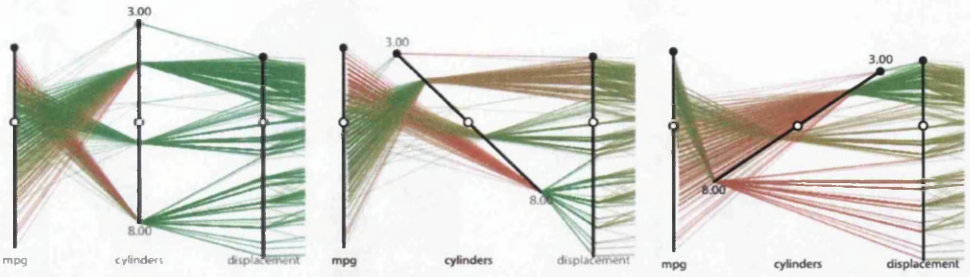
The situation with more than two axes is more difficult to interpret. Small rotations in each preceding axis can result in a large rotation for axes in the center of the plot. We address this through interactions as described in the following section, but initial axis order still plays an important role. While we do not address in this work the problem of an appropriate axis order, we support axis re-arrangement through user interaction using a simple drag-and-drop metaphor.

A.3 Interactions

As a static image, a force-directed PCP shows advantages over a standard PCP. Through interacting with the system, a user can gain an even better understanding of the underlying data. In this section, we present three new interaction techniques for force-directed parallel coordinates and give brief details of their implementation in our system.

A.3.1 Cutting

Brushing on a PCP can be a filtering operation as well as a selection operation: polylines can be removed instead of simply colored. The natural analogue of this operation on a force-directed PCP is cutting. Selection on an axis can be used to determine springs to cut. These springs (and the complete polyline of which they form part) are then removed from the physical system, as shown in Figure A.4. To return to our bridge metaphor, this is equivalent to cutting cables, and will result in a change in the forces on the axis and hence possibly a rotation. However, in our system, since the whole polyline (multiple springs) is removed, more than one axis may change orientation. These changes give information about both the data remaining and the data removed, and the animated transition in axis orientation may improve the perception of changes between the states [HR07].



(a) Before (b) Swinging counter-clockwise (c) Swinging clockwise

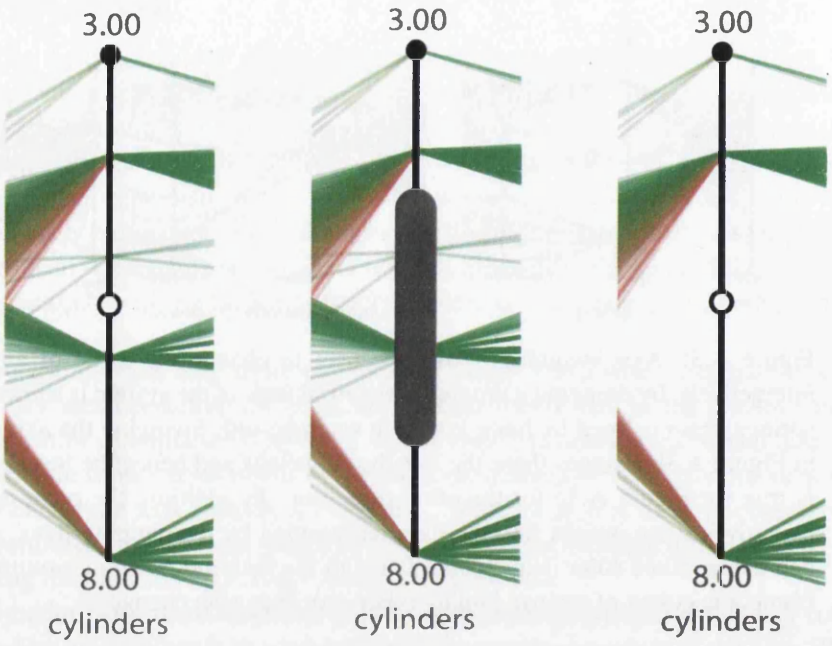
Figure A.3: Axis swinging allows the user to change the angle of any axis in the system interactively, by dragging it around. The initial state of the system is shown in Figure A.3a, with springs again colored by force low-high as green-red. Swinging the axis counter-clockwise as in Figure A.3b changes these the lengths of springs and hence the forces they exert. The same is true for Figure A.3c for the other direction. In addition, the process of manipulating the axis gives some insight into the data represented by overlapping lines, as they change angle and color. Since color is mapped relative to the maximum and minimum forces in the current frame, the colors of springs joining other axes may also change.

Our implementation for this interaction is straightforward. First, the transformed start and end values for the selection are calculated. Next, springs in the system with a start or end point in this range are identified, and by following the chain of springs across axes in both directions, a list of springs to remove is compiled. Finally, these springs are all removed from the system.

A.3.2 Axis Swinging

The most natural interaction with any spring system is to allow pushing or pulling directly. Most systems, once released, will return to their original state. In our system, pushing and pulling an axis to rotate manually accomplishes two tasks. First, it gives the user a better indication of the positions of springs connected to that axis, since they will appear to slide over each other (and, in our implementation, perhaps change color, if the forces change). This is similar in some respects to the idea of jittering in scatterplots. Second, it gives the opportunity for the system to return to a different state: if, as in the strong negative correlation examples in Figure A.2, there is more than one stable state for the system, the user can force the system out of the local minima into the other state.

There are two cases within our implementation of axis swinging: with the physics engine active and with it paused. With the engine running, we simply calculate the new angle for the axis the user wishes to swing, and force it to that position. Releasing the mouse button releases the axis back into the system at its new position. With the physics system paused, we fix all the other axes in their current orientation, and update only the positions of springs connected to the axis under manipulation. These two cases were suggested by informal user feedback on our system: having the ability to force axes to angles with the system in motion was perceived as a very different operation to manipulating the system with all axes at rest, but both are considered



(a) Before (b) During selection (c) After

Figure A.4: Cutting is a filtering operation in our system that works by removing springs within the brushed region. The related springs linking other axes are also removed, which may trigger a shift in orientation of the axis. Here, springs are colored based on the force they exert, from low (green) to high (red). Note that the cutting operation does not result in an instant change in forces on the remaining springs, though the evolution of the system with the selected springs removed may eventually do so.

useful.

A.3.3 Axis Pinning

In exploring the system, changes made to on axis can potentially affect another non-adjacent axis in an unexpected way. Perhaps a cutting operation results in an axis settling at a right-angle to its neighboring axis — the worst possible case for occlusion in our system. While we do not tackle the axis re-ordering problem directly, we offer a partial solution to this problem through interaction by giving users the ability to pin or freeze an axis in position.

This action splits the physical system into two separate systems at the point of the pinned axis. It can be repeated as many times as required to form smaller and smaller systems, giving the ability to examine purely local effects between adjacent axes without changing orientations elsewhere. Implementation is simple: the end points of the rod are fixed, which fixes the end points of connected springs also. Pinned axes are shown visually as translucent.

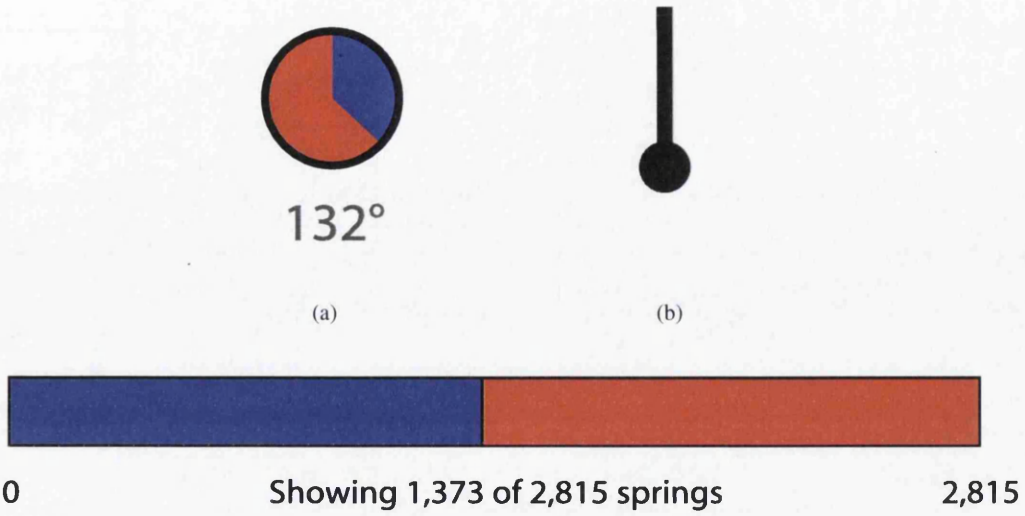


Figure A.5: Instrumentation for force-directed PCPs. (a) shows the angle through which an axis has rotated, (b) denotes the low-value end of an axis and (c) indicates the number of springs currently shown as a proportion of the original spring count.

A.4 Example

Having introduced the technique and interactions, in this section we present an example of using force-directed PCPs to examine a multidimensional data set. The focus of this example is less on the insights gained than on how the techniques described above can be used to explore the data set.

To facilitate comparison with existing work, we use the cars dataset from the American Statistical Association Data Exposition of 1983. As with all interaction techniques, they are difficult to fully illustrate on paper, therefore we highly encourage the reader to view the accompanying video. To help the user track the results of axis manipulation and cutting operations, we instrument the display of our plots with an angle tracker (Figure A.5a) and spring count (Figure A.5b), and denote the lower end of an axis as in Figure A.5c.

A.4.1 Cars

The cars data set consists of 406 rows of 9 dimensional data on cars tested by the Consumer Reports magazine between 1971 and 1983. Of the 9 variables per car — make and model, fuel economy, cylinders, displacement, horsepower, weight, acceleration, model year and origin — we exclude make and model from our consideration. While PCP-like techniques can deal with categorical data, either by simply mapping each category to a numerical value or by other extensions [KBH06], it poses a different set of problems to force-directed parallel coordinates, because the arithmetic average used to position the pivots is meaningless for a category variable

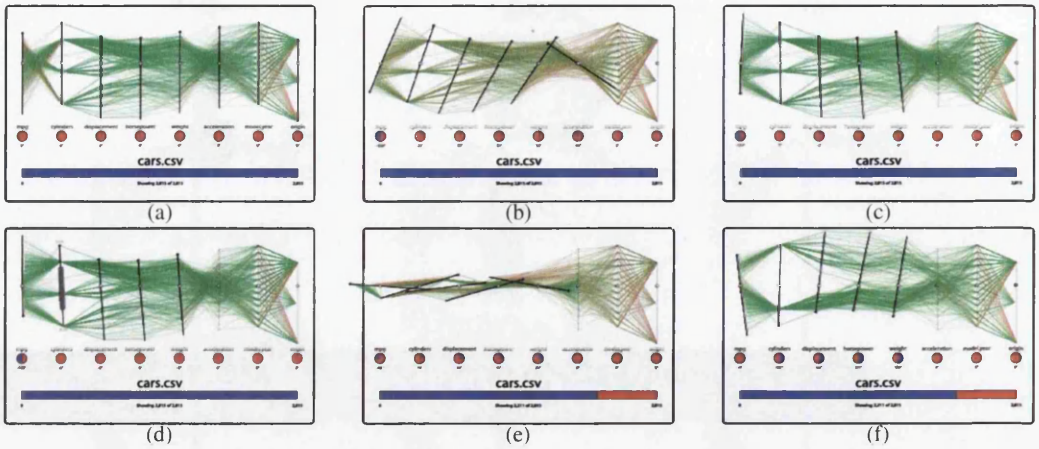


Figure A.6: Analysis of the cars data set using a force-directed PCP. (a) shows the initial state, (b) the stable state (c) the result of swinging and pinning interactions, (d) a cutting operation (e) a transitional state before arriving to (f) the stable state.

mapped to a number range. For the origin dimension, which falls into this category, we avoid the problem by pinning the axis in our tool.

The starting point for our analysis is shown in Figure A.6a. Axes are arranged in the default order, and some correlations are visible: MPG and cylinders seem to show some negative correlation, and displacement and horsepower are also correlated. Allowing the axes to swing freely gives a state as in Figure A.6b. The MPG axis has inverted itself automatically. Its new orientation shows positive correlation with cylinders, but the color scheme is largely dictated by the springs in the weight-acceleration-model.year set. In fact, the acceleration axis settles at an angle approaching 90 degrees to its neighboring axes — the worst possible case for occlusion.

Here, we begin to interact with the data. We can swing the acceleration and model.year axes to an upright position and pin them there (Figure A.6c). This gives a stable state for the system, but we decide to investigate further — we filter on the cylinder axis to remove cars with $4 < \textit{cylinders} < 8$ by cutting (Figure A.6d). This changes the forces on the weight axis sufficiently for it to invert, and this in turn triggers inversion of the axes to the left of it, passing through the state in Figure A.6e before arriving in the state described in Figure A.6f, which shows the remaining data split into two groups with a minimal amount of overlap. Our final conclusions from this state — that, looking at cars with less than 4 or eight cylinders, mpg is inversely correlated to cylinders which is directly correlated to displacement, horsepower and weight. The process followed helps provide some direct understanding of these relationships.

A.5 User Study

We conducted a usability evaluation of our system using the same methodology as Claessen and van Wijk [CvW11], with a total of 13 participants between the ages of 18 and 34. The

gender split was 12 male to 1 female, all participants had normal color vision and 10 were already familiar with parallel coordinate plots.

Our experiment consisted of six phases: an introduction, discussing the ideas behind parallel coordinates, the differences between PCPs and FDPCPs and the purpose of the experiment; a demonstration of the new interaction techniques using the cars data set; a user exercise on the Iris data set; a second demonstration using this set; user exploration on a previously-unseen data set; and finally a survey to gather opinions and comments on this exploration.

In the user exercise, participants were asked to answer a set of seven questions:

1. What is the range of values of attribute Sepal width?
2. Is there a correlation between Sepal length and Sepal width?
3. Which value occurs most often for Sepal width?
4. What is the average value for Sepal width?
5. Is there a correlation between Sepal length and Petal length?
6. Is there a correlation between Sepal length and Petal width?
7. Which attribute(s) lend themselves for classification?

It should be noted that question 3 (determining the mode for sepal width) is not easily answerable using FDPCPs - this was included as a test of their understanding of the plots, and discussed during the second demonstration. Question 3 aside, participants were able to answer all questions correctly. The user exploration phase was performed as a 'think-aloud' exercise: participants explored a data set of world food prices since 1990 — overall index for food, and separate breakdowns for dairy, cereal, meat, oils and sugar, as shown in Figure A.7 — while describing their use of interactions and the expected result. The role of the experimenter was to watch and record their statements and activities. This data set includes a number of interesting patterns, such as the 2007–2008 world food price crisis.

During the experiment, we observed a number of approaches to the exploration task. Several participants preferred to work primarily with the physical system turned off, and enable it only to see the affect of significant cutting or axis rotations on other axes. Color was used as a cue to identify data points of possible interest for further exploration. Brushing was used more often than cutting: participants recognized that cutting in the current system is an irreversible operation, and preferred to explore thoroughly before removing data in this fashion.

The survey on completion of the exploration task contained a number of statements on which participants could offer an opinion using a 5-point Likert scale. The results of this survey are shown in Table A.1.

In the comments sections, participants considered the strength of FDPCPs to be in identification of correlations and trends (mentioned explicitly by 6 participants). Other comments concerned the usefulness of interactions - participants felt that the natural movement of axes in the system was less useful than rotation driven by the user. The ability to toggle the physical system on and off was also mentioned as a strength: while participants were in general agreement that the physical model and interactions helped gain understanding, these interactions were complicated if the system was constantly in motion.

	++	+	o	-	- -
FDPC					
– system is easy to use	4	5	2	2	
– technique is easy to understand	6	6	1		
Helped to understand the data					
– Watching movement of axes	3	6	3	1	
– Moving axes manually	6	5	2		
– Cutting springs	10	2	1		
– Pinning axes	6	6	1		
– Marking springs	6	3		3	1
– Coloring springs by length	5	2	1	3	2
Added value					
– moving over fixed axes	4	6	3		
– FDPCPs over PCPs	6	6	1		

Table A.1: Summary of survey responses from user study, with the encodings Strongly agree (++), Slightly Agree (+), Neutral (o), Slightly Disagree (-), Strongly Disagree (- -). Participants were generally positive about the new interactions with the plots.

With respect to weaknesses, we received much useful feedback concerning elements of the interface. Participants wanted the ability to adjust pivot points and disliked that cutting springs was a one-way operation with no way to reverse the process without resetting the system. The interactions required very precise mouse input, and this interfered with the naturalness of the exploration.

In summary, the participants were positive about both the technique and the prototype system, and suggested a large number of useful improvements to the software.

A.6 Discussion

This paper introduces force-directed parallel coordinates, a model and set of related interactions for multidimensional data based on a physical interpretation of parallel coordinate plots. We have defined and demonstrated cutting, axis swinging, and axis pinning for these plots, and our user study indicates that users feel that these interaction techniques are of benefit in exploring a data set. While these techniques are described here, we strongly recommend that the reader view the accompanying video to see them in operation.

A.6.1 Limitations

While these examples demonstrate the usefulness of our interaction techniques, force-directed PCPs have some limitations. The initial state of the system is dependent on axis ordering, axis scaling and pivot position. Given an unfavorable axis ordering, these interaction techniques may reveal little concerning the data. However this is true for parallel coordinates visualizations in general and not specific to our case only. While our choices of pivot position and axis

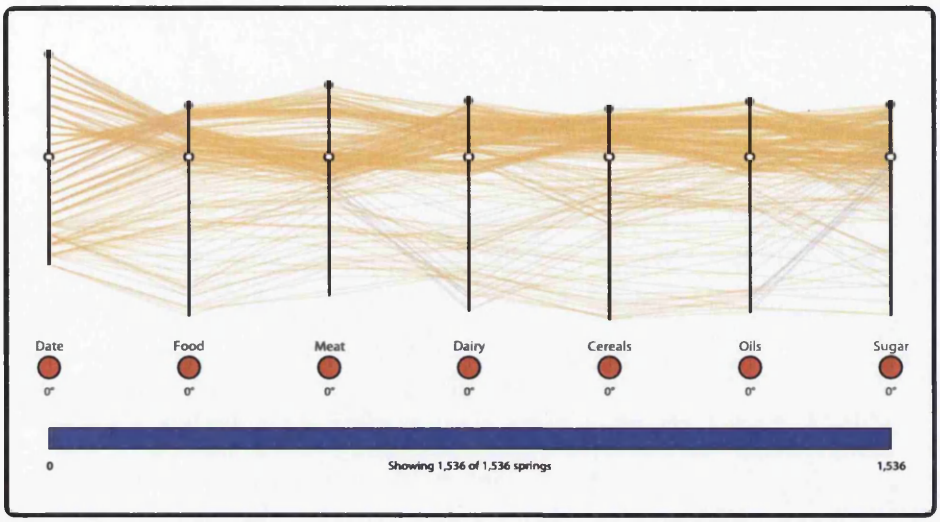


Figure A.7: FAO Food Price Indices since 1990 data set, as used for the data exploration phase of the user study described in Section A.5, springs colored by force low-high as orange-blue. While some patterns are visible in this image, such as the decline in food prices in the early 2000s, other, more complex patterns can be discovered through interaction with the plot.

scaling are made to produce initially stable states and minimize occlusion, they may not be the best choice for other data sets with different characteristics.

The current implementation is also unsuited to large data sets, due to both axis ordering as above and also computational complexity. The chief consequence of the change in complexity is that interactivity is quickly lost: cutting and pinning can act on a paused system, but axis swinging is far less useful as an operation when each movement requires lengthy computation before an update.

A.6.2 Future Work

There is much interesting future work to be completed on this system. Integration of our system with other existing techniques to address its limitations in areas such as axis re-ordering would be beneficial, as would interactions like angular cutting and perhaps other physical interactions, such as user-adjustment of pivot position by sliding axes up or down. Our implementation of the physical system could be optimized by performing many of the mass-spring calculations on the GPU [GW05], which would enable consideration of larger data sets. It may be possible, through careful configuration of the system, to mimic statistic measures in a physical fashion. From a computational perspective this would be inefficient, but having a physical representation of, say, principal component analysis expressed as a set of axis rotations could be valuable to user understanding. We are also currently engaged in a larger scale, crowdsourced user study [HB10] comparing force-directed PCPs to standard PCPs, which should help us to quantify possible benefits for a number of tasks and data sets.

Bibliography

[ABK98] M. Ankerst, S. Berchtold, and D.A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *infovis*, page 52. Published by the IEEE Computer Society, 1998.

[AdOL04] Almir Olivette Artero, Maria Cristina Ferreira de Oliveira, and Haim Levkowitz. Uncovering Clusters in Crowded Parallel Coordinates Visualizations. In *IEEE Information Visualization Conference*, pages 81–88. IEEE Computer Society, 2004.

[ADV] ADVANCED VISUAL SYSTEMS INC. OpenViz. 300 Fifth Avenue,Waltham, MA 02451. http://www.avs.com.

[Bau32] Wolf Graf Baudissin. *Othello, der Mohr von Venedig*. [edited by R Wenig for Project Gutenberg], *http://gutenberg.spiegel.de/buch/2185/1*, 1832.

[BBM+06] Rajiv Badi, Soonil Bae, J. Michael Moore, Konstantinos Meintanis, Anna Zacchi, Haowei Hsieh, Frank Shipman, and Catherine C. Marshall. Recognizing User Interest And Document Value From Reading And Organizing Activities In Document Triage. In *Proceedings of International conference on intelligent user interfaces*, pages 218–225. ACM Press, 2006.

[BBP08a] J. Blaas, C. Botha, and F. Post. Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. — *IEEE Transactions on Visualization and Computer Graphics*, pages 1436–1451, 2008.

[BBP08b] Jorik Blaas, Charl P. Botha, and Frits H. Post. Extensions of Parallel Coordinates for Interactive Exploration of Large Multi-Timepoint Data Sets. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1436–1451, 2008.

[BGN08] B.Scott., G.Carl., and N.Miguel. Seeing Things in the Clouds: The Effect of Visual Features on Tag Cloud Selections. In *HT '08: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 193–202, New York, NY, USA, 2008. ACM.

[BH06] Cindy Brewer and Mark Harrower. Colorbrewer, May 2006. `http://colorbrewer.org/`.

[BHvW00] Mark Bruls, Kees Huizing, and Jarke J. van Wijk. Squarified Treemaps. In *Proceedings Joint Eurographics/IEEE TVCG symposium Visualization*, pages 33–42, 2000.

[BL07] George Buchanan and Fernando Loizides. Investigating Document Triage On Paper And Electronic Media. In *Proceedings of European Conference on Research and advanced Technology for Digital Libraries*, pages 416–427, 2007.

[Bol85] Hanno Bolte. *Othello: Englisch-Deutsch : William Shakespeare.* Herausgegeben von Dieter Hamblockk, 1985.

[Bru47] Karl Brunner. *William Shakespeare, Othello, der Mohr von Venedig. Englischer Text mit deutscher Übersetzung nach Ludwig Tieck.* Britisch-Amerikanische Bibliothek, 1947.

[BT04] Baudel and Thomas. Browsing Through an Information Visualization Design Space. In *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, volume 2 of *Demonstrations*, pages 765–766, 2004.

[Car91] Daniel Carr. Looking at Large Data Sets Using Binned Data Plots. *Computing and Graphics in Statistics,ed. by Buja, A., Turkey,P.A*, pages 7–39, 1991.

[Car98] Carnegie Mellon University. DocuScope: Computer-aided Rhetorical Analysis, 1998. `http://www.cmu.edu/hss/english/research/docuscope.html`, Last Access Date: 2013-1-16.

[CBFK93] C. Cool, N. J. Belkin, O. Frieder, and P. Kantor. Characteristics of Texts Affecting Relevance Judgments. In *In 14th National Online Meeting*, pages 77–84, 1993.

[CBW09] Christopher Collins, Fernada B.Viegas, and Martin Wattenberg. Parallel Tag Clouds to Explore and Analyze Facted Text Corpora. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 91–98. IEEE Computer Society, 2009.

[CCP09] Christopher Collins, M. Sheelagh T. Carpendale, and Gerald Penn. DocuBurst: Visualizing Document Content using Language Structure. *Computer Graphics Forum*, 28(3):1039–1046, 2009.

[CFT12] Tom Cheesman, Kevin Flanagan, and Stephan Thiel. Translation Array Prototype , 2012. *delightedbeauty.org/vvv.*

[Cla08] Jeff Clark. Document contrast diagrams, 2008. `http://neoformix.com/2008/DocumentContrastDiagrams.html`, Last Access Date: 2011-2-18.

[CM85] W. S. Cleveland and R. McGill. Graphical Perception and Graphical Methods for Analyzing Scientific Data. *Science*, 229(4716):828–833, 1985.

[CN02] Stuart K. Card and David Nation. Degree-of-Interest Trees: A Component of an Attention-Reactive User Interface. In *working conference on advanced visual interfaces (AVI)*, pages 231–245, 2002.

[CtVVVPT11] Tom Cheesman and the Version Variation Visualization Project Team. 'Translation Sorting: Eddy and Viv in Translation Arrays, 2011. `http://www.scribd.com/doc/101114673/ Eddy-and-Viv`.

[CvW11] Jarry H.T. Claessen and Jarke J. van Wijk. Flexible linked axes for multivariate data visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2310 –2316, dec. 2011.

[CWG11] Michael Correll, Michael Witmore, and Michael Gleicher. Exploring Collections of Tagged Text for Literary Scholarship. *Computer Graphics Forum*, 30(3):731–740, 2011.

[Dav92] Mark L. Davison. *Multidimensional Scaling*. Robert E. Krieger Publishing Co. Inc., Malabar, FL, 1992.

[DBETT94] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry-Theory and Application*, 4(5):235–282, 1994.

[DDF+90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41, 1990.

[DK10a] A. Dasgupta and R. Kosara. Pargnostics: Screen-Space Metrics for Parallel Coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1017–1026, 2010.

[DK10b] Aritra Dasgupta and Robert Kosara. Pargnostics: Screen-Space Metrics for Parallel Coordinates. *IEEE Transaction on Visualization and Computer Graphics*, 16(6):1017–1026, 2010.

[dOL03] Maria Cristina Ferreira de Oliveira and Haim Levkowitz. From Visual Data Exploration to Visual Data Mining: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):378–394, 2003.

[ED06] Geoffrey Ellis and Alan J. Dix. Enabling Automatic Clutter Reduction in Parallel Coordinate Plots. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):717–724, 2006.

[ED07] Geoffrey Ellis and Alan Dix. A Taxonomy of Clutter Reduction for Information Visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007.

[Eng76] Balz Engler. *Othello: Englisch-deutsche Studienausgabe*. Munich: Franke, 1976.

[FKLT10] David Feng, Lester Kwock, Yueh Lee, and Russell M. Taylor. Matching Visual Saliency to Confidence in Plots of Uncertain Data. *IEEE Transaction on Visualization and Computer Graphics*, 16(6):980–989, 2010.

[Fla09] Richard Flatter. *Othello der Mohr von Venedig*. Theater-Verlag Desch, Munich, 2009.

[Fod02] Imola Fodor. A Survey of Dimension Reduction Techniques. Technical report, Technical report, 2002.

[FR91] T.M.J. Fruchterman and E.M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[Fri99] Erich Fried. *Hamlet und Othello*. Berlin: Verlag Klaus Wagenbach, 1999.

[FWR99] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical Parallel Coordinates for Exploration of Large Datasets. In *IEEE Visualization*, pages 43–50, 1999.

[Gil08] O. T. Gilson. *An Ontological Approach to Information Visualization*. Phd thesis, Department of Computer Science, Swansea University, UK, 2008.

[GJL+09] Edward Grundy, Mark W. Jones, Robert S. Laramee, Rory P. Wilson, and Emily L. C. Shepard. Visualisation of Sensor Data from Animal Movement. *Computer Graphics Forum*, 28(3):815–822, 2009.

[GLC+11] Zhao Geng, Robert S. Laramee, Tom Cheesman, Alison Ehrmann, and David M. Berry. Visualizing translation variation: Shakespeare's othello. In *Advances in Visual Computing, Lecture Notes in Computer Science LNCS, Volume 6938 (Proceedings of the 7th International Symposium on Visual Computing (ISVC)*, volume 6938 of *Lecture Notes in Computer Science*, pages 653–663. Springer, 2011.

[GLF+12] Zhao Geng, Robert S. Laramee, Kevin Flanagan, Stephan Thiel, and Tom Cheesman. Visual Analysis of Segment Variation of German Translations of Shakespeare's Othello, 2012. Technical Report, Department of Computer Science, University of Wales, Swansea, UK.

[GLLB11] Zhao Geng, Robert S. Laramee, Fernando Loizides, and George Buchanan. Visual Analysis of Document Triage Data. In *International Conference on Information Visualization Theory and Application (IVAPP), Vilamoura, Algarve, Portugal, March 5-7*, pages 151–163. SciTePress, 2011.

[GPL+11] Zhao Geng, ZhenMin Peng, Robert S. Laramee, Rick Walker, and Jonathan Roberts. Angular Histograms: Frequency Based Visualizations For Large, High-Dimensional Data. *IEEE Transaction on Visualization and Computer Graphics*, 17(6):2572 – 2580, 2011.

[GSC+12] Zhao Geng, Robert S.Laramee, Tom Cheesman, Andrew Rothwell, David M. Berry, and Alison Ehrmann. 'Visualizing Translation Variation of Othello: A Survey of Text Visualization and Analysis Tools, 2012. Technical Report, Department of Computer Science, University of Wales, Swansea, UK.

[GW05] J. Georgii and R. Westermann. Mass-spring systems on the GPU. *Simulation Modelling Practice and Theory*, 13(8):693–702, 2005.

[GWL12] Zhao Geng, James Walker, and Robert S. Laramee. Markov chain driven multi-dimensional visual pattern analysis with parallel coordinates. In *Proceedings of Vision, Modeling, and Visualization (VMV)*, pages 191 – 198, 2012.

[Ham06] Maik Hamburger. Translating and Copyright, 2006. In Hoenselaars, T., ed., Shakespeare and the Language of Translation, London: Arden, pp. 148-166.

[Han05] Sameh Hanna. *Othello* in Egypt: Translation and the (Un)making of National Identity, 2005. In J.House, M. Rosario Martn Ruano and N.Baumgarten, eds, Translation and the Construction of Identity. IATIS Yearbook 2005, Manchester: St.Jerome: 109-128.

[HB10] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 203–212. ACM, 2010.

[HGM+97] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. DNA visual and analytic data mining. In *Proceedings of the 8th conference on Visualization'97*, pages 437–ff. IEEE Computer Society Press, 1997.

[HH00] William W. Hargrove and Forrest M. Hoffman. An Analytical Assessment Tool for Predicting Changes in a Species Distribution Map Following Changes in Environmental Conditions. In *4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs*, pages 11 – 18, 2000.

[HHWN02] Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[HK05] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[HLD02a] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 127–130. IEEE, 2002.

[HLD02b] Helwig Hauser, Florian Ledermann, and Helmut Doleisch. Angular Brushing of Extended Parallel Coordinates. In *Proceedings of IEEE Symposium on Information Visualization*, pages 127–130. IEEE Computer Society, 2002.

[HMM00] I. Herman, G. Melançon, and M.S. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.

[HMSA08] Jeffrey Heer, Jock D. Mackinlay, Chris Stolte, and Maneesh Agrawala. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, 2008.

[Hol06] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, pages 741–748, 2006.

[Hom11] KDnuggets Home. Visualization Software, Feb 2011. `http://www.kdnuggets.com/software/visualization.html`, Last Access Date: 2011-2-18.

[HR07] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, pages 1240–1247, 2007.

[HS03] Blinn H. and W.G. Schmidt. *Shakespeare deutsch: Bibliographie der bersetzungen und Bearbeitungen.* 2003.

[HW04] Jonathan Hope and Michael Witmore. The Very Large Textual Object: A Prosthetic Reading of Shakespeare. *Early Modern Literary Studies*, 9(3):1–36, 2004.

[ID90a] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization'90*, pages 361–378. IEEE Computer Society Press, 1990.

[ID90b] Alfred Inselberg and B. Dimsdale. Parallel Coordinates: A Tool for Visualizing Multi-dimensional Geometry. In *Proceedings of IEEE Visualization*, pages 361–378, 1990.

[Inf11] InfoChimps. Daily 1970-2010 Open, Close, Hi, Low and Volume (NYSE exchange), 2011. `http://www.infochimps.com/datasets/`, NASDAQ Exchange Daily 1970-2010 Open, Close, High, Low and Volume, Last Access Date: 2011-3-16.

[Ins08] A. Inselberg. *Parallel coordinates: visual multidimensional geometry and its applications*. Springer-Verlag New York Inc, 2008.

[Ins09] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009.

[J.-11] J.-K. Chou, C.-K. Yang. PaperVis: Literature Review Made Easy. *Computer Graphics Forum*, 30(1):721–730, 2011.

[JC08] J. Johansson and M. Cooper. A screen space quality method for data abstraction. In *Computer Graphics Forum*, volume 27, pages 1039–1046. Wiley Online Library, 2008.

[JLJC05] Jimmy Johansson, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing Structure within Clustered Parallel Coordinates Displays. In *IEEE Information Visualization Conference*, pages 17–25. IEEE Computer Society, 2005.

[Jon09] Jonathan Feinberg. Wordle: Beautiful Word Clouds, 2009. `http://www.wordle.net/`, Last Access Date: 2011-2-18.

[JS91] B. Johnson and Ben Shneiderman. Tree Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. In *IEEE Visualization*, pages 284–291, 1991.

[JSS00] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2):207–227, 2000.

[JWS+05] D. Jonker, W. Wright, D. Schroh, P. Proulx, and B. Cort. Information Triage With Trist. In *In Proceedings Intelligence Analysis*, pages 1–6, 2005.

[KBH04] Robert Kosara, Fabian Bendix, and Helwig Hauser. TimeHistograms for Large, Time-Dependent Data. In *Joint EUROGRAPHICS-IEEE TCVG Symposium on Visualization*, pages 45–54, 340. Eurographics Association, 2004.

[KBH06] R. Kosara, F. Bendix, and H. Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, pages 558–568, 2006.

[Kei02] Daniel A. Keim. Information Visualization and Visual Data Mining. *IEEE-TVCG: IEEE Transactions on Visualization and Computer Graphics*, 8:1–8, 2002.

[KHG03a] R. Kosara, H. Hauser, and D. Gresh. An interaction view on information visualization. *State-of-the-Art Report. Proceedings of EUROGRAPHICS*, 2003.

[KHG03b] Robert Kosara, Helwig Hauser, and Donna L. Greshn. An Interaction View on Information Visualization. In *EUROGRAPHICS*, pages 123–137, 2003.

[KJ02] Jaana Kekäläinen and Kalervo Järvelin. Using Graded Relevance Assessments In IR Evaluation. *Journal of the American Society for Information Science*, 53(13):1120–1129, 2002.

[KK96] Daniel A. Keim and Hans-Peter Kriegel. Visualization techniques for Mining Large Databases: A Comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):923–938, 1996.

[KKZ09] Hans-Peter Kriegel, Peer Koger, and Arthur Zimek. Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering. *ACM Transaction on Knowledge Discovery from Data*, 3(1), 2009.

[KLKS10] Kyle Koh, Bongshin Lee, Bo Hyoung Kim, and Jinwook Seo. ManiWordle: Providing Flexible Control over Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1190–1197, 2010.

[Kob01] Alfred Kobsa. An Empirical Comparison of Three Commercial Information Visualization Systems. In *In proceedings of InfoVis 2001, IEEE Symposium on Information Visualization, San Diego, CA*, pages 123–130, 2001.

[Kob04] Alfred Kobsa. User Experiments with Tree Visualization Systems. In *IEEE Information Visualization*, pages 9–16. IEEE Computer Society, 2004.

[Kos11] R. Kosara. Indirect multi-touch interaction for brushing in parallel coordinates. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7868, page 7, 2011.

[Lau78] Horse Laube. *Othello Der Mohr von Venedig überset und bearbeitet von Horst Laube*. Frankfurt am Main: Verlag der Autoren, 1978.

[Lau96] Erich Selbmann Lauterbach. *Othello, der Mohr von Venedig*. Henschel Schauspiel Theaterverlag Berlin, 1996.

[LB09] Fernando Loizides and George Buchanan. An Empirical Study of User Navigation during Document Triage. In *Proceedings of Research and Advanced Technology for Digital Libraries, 13th European Conference*, volume 5714, pages 138–149. Springer, 2009.

[LCS97] Dik Lun Lee, Huei Chuang, and Kent E. Seamons. Document Ranking and the Vector-Space Model. *IEEE Software*, 14(2):67–75, 1997.

[LMDK07] Thomas Landauer, Danielle McNamara, Simon Dennis, and Walter Kintsch. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

[LMvW10] Jing Li, Jean-Bernard Martens, and Jarke J. van Wijk. Judging correlation from scatterplots and parallel coordinate plots. *Information Visualization*, 9(1):13–30, 2010.

[LPL+08] Emily L.C.Shepard, Rory P.Wilson, Nikolai Liebsch, Flavio Quintana, Agustina Gomez Laich, and Klaus Lucke. Flexible paddle sheds new light on speed: a novel method for the remote measurement of swim speed in aquatic animals. *Endang Species Res*, 4(6):157–164, 2008.

[LRKC10] Bongshin Lee, Nathalie Henry Riche, Amy K. Karlson, and M. Sheelagh T. Carpendale. SparkClouds: Visualizing Trends in Tag Clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189, 2010.

[M. 05] M. Taylor. *TOPCAT - Tool for OPerations on Catalogues And Tables Version 3.4-3*. Starlink development, 2005.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. LeCam and J. Neyman, editors, *Proc. of the 5th Berkeley Symp. on Mathematics Statistics and Probability*, 1967.

[Meh06] Chirag Mehta. Tagline Generator - Timeline-based Tag Clouds, 2006. `http://chir.ag/projects/tagline/`, Last Access Date: 2011-2-18.

[MHDG11] Philipp Muigg, Markus Hadwiger, Helmut Doleisch, and Eduard Gröller. Visual Coherence for Large-Scale Line-Plot Visualizations. *Computer Graphics Forum*, 30(3):643–652, 2011.

[Mic07] Microsoft office. *Microsoft Office Excel 2007 product guide*, 2007.

[MS92] Marsh and Shaun. The Interactive Matrix Chart. *ACM SIGCHI Bulletin*, 24(4):32–38, 1992.

[MS08] Theus Martin and Urbanek Simon. *Interactive Graphics for Data Analysis: Principles and Examples (Computer Science and Data Analysis)*. Chapman & Hall/CRC, 2008.

[MW95] A.R. Martin and M.O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th Conference on Visualization'95*, page 271. IEEE Computer Society, 1995.

[NH06] Matej Novotny and Helwig Hauser. Outlier-Preserving Focus+Context Visualization in Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006.

[Pal02] W. Bradford Paley. TextArc: An Alternative Way to View Text, 2002. `http://www.textarc.org/`, Last Access Date: 2011-2-18.

[PWR04] W. Peng, M.O. Ward, and E.A. Rundensteiner. Clutter reduction in multidimensional data visualization using dimension reordering. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 89–96. IEEE, 2004.

[Rab89] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[RHM+12] Christian Rohrdantz, Michael Hund, Thomas Mayer, Bernhard Wälchli, and Daniel A. Keim. The world's languages explorer: Visual analysis of language features in genealogical and areal contexts. *Computer Graphics Forum*, 31(3):935–944, 2012.

[RK08] Oliver Ruebel and Wu K. High Performance Multivariate Visual Data Exploration for Extremely Large Data. Lawrence Berkeley National Laboratory, Barkeley, CA 94720, USA, 2008.

[RTT03] José Fernando Rodrigues, Agma J. M. Traina, and Caetano Traina. Frequency Plot and Relevance Plot to Enhance Visual Data Exploration. In *SIBGRAPI*, pages 117–124. IEEE Computer Society, 2003.

[Rud63] Rudolf Alexander Schröder. *Shakespeare deutsch.* Berlin Frankfurt Suhrkamp, 1963.

[RVA04] Martin Rajman, Martin Vesely, and Pierre Andrews. State of the Art, Evaluation and Recommendations Regarding Document Processing and Visualization Techniques, 2004. `http://arxiv.org/abs/cs/0412114`, Last Access Date: 2011-2-18.

[Seh09] Sehnaz Tahir Gürcaglar. Retranslation, 2009. In M. Baker and G. Saldanha (eds.), Encyclopedia of Translation Studies. Abingdon and New York: Routledge, pages 232-36.

[Shn92] Ben Shneiderman. Tree Visualization With Treemaps: a 2-d Space-filling Approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[Shn96] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[Shn03] B. Shneiderman. Why Not Make Interfaces Better than 3D Reality? *IEEE Computer Graphics and Applications*, 23(6):12–15, 2003.

[Sii00] H. Siirtola. Direct manipulation of parallel coordinates. In *Information Visualization, 2000. Proceedings. IEEE International Conference on*, pages 373–378. IEEE, 2000.

[Sil86] B. W. Silverman. Kernel Density Estimation Technique for Statistics and Data Analysis. In *Monographs on statistics and applied probability*, volume 26. Chapman and Hall, 1986.

[SJWS02] A. Spink, B. J. Jansen, D. Wolfram, and T. Saracevic. From e-sex to e-commerce: Web search changes. *IEEE Computer*, 35(3):107–109, 2002.

[Spe01] Robert Spence. *Information Visualization.* Adisson-Wesley, 2001.

[Spe07] Robert Spencer. *Information Visualization: Design for Interaction.* Pearson Education Limited, 2007.

[SR06] H. Siirtola and K.J. Räihä. Interacting with parallel coordinates. *Interacting with Computers*, 18(6):1278–1309, 2006.

[SSS+12] Hendrik Strobelt, Marc Spicker, Andreas Stoffel, Daniel A. Keim, and Oliver Deussen. Rolled-out wordles: A heuristic method for overlap removal of 2D data representatives. *Computer Graphics Forum*, 31(3):1135–1144, 2012.

[Ste08] Daniel Steinbock. TagCrowd: Joining the Crowd Together , 2008. `http://tagcrowd.com/`, Last Access Date: 2011-2-18.

[SVM+99] Marc M. Sebrechts, Joanna Vasilakis, Michael S. Miller, John V. Cugini, and Sharon J. Laskowski. Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces. In *In Proceedings of ACM SIGIR, New York*, pages 3–10. ACM Press, 1999.

[TAE+09] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnork, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 59–66. IEEE, 2009.

[TAvHS06] C. Tominski, J. Abello, F. van Ham, and H. Schumann. Fisheye Tree Views and Lenses for Graph Visualization. In *Information Visualization, 2006. IV 2006. Tenth International Conference on*, pages 17–24. IEEE, 2006.

[TC09] Alfredo Raúl Teyseyre and Marcelo R. Campo. An Overview of 3D Software Visualization. *IEEE Trans. Vis. Comput. Graph*, 15(1):87–105, 2009.

[TFH11] Cagatay Turkay, Peter Filzmoser, and Helwig Hauser. Brushing dimensions - A dual visual analysis model for high-dimensional data. *IEEE Transaction on Visualization and Computer Graphics*, 17(12):2591–2599, 2011.

[The02] Martin Theus. Interactive Data Visualization Using Mondrian. *Journal of Statistical Software*, 7(11):1–9, 2002.

[Thi06] Stephan Thiel. Understanding Shakespeare, 2006. `http://www.understanding-shakespeare.com/`, Last Access Date: 2013-1-16.

[T.J11] T.J. Jankun-Kelly and David Wilson and Andrew S. Stamps and Josh Franck and Jeffry Carver and J. Edward Swan II. Visual Analysis for Textual Relationships in Digital Forensics Evidence. *Information Visualization, Special Issue on VizSec 2009*, 10(2):134–144, 2011.

[UTH06] Antony Unwin, Martin Theus, and Heike Hofmann. *Graphics of Large Datasets: Visualizing a Million (Statistics and Computing)*. Springer, 2006.

[Ver67] L. Verlet. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Physical Review*, 159(1):98–103, 1967.

[vHWV09] Frank van Ham, Martin Wattenberg, and Fernanda B. Viégas. Mapping Text with Phrase Nets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1169–1176, 2009.

[VWF09] Fernanda B. Viegas, Martin Wattenberg, and Jonathan Feinberg. Participatory Visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1137–1144, 2009.

[VWvH+07] Fernanda B. Vigas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. ManyEyes: A Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.

[War94a] Matthew O. Ward. XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proceedings of IEEE on Visualization*, pages 326–336. IEEE Computer Society Press, 1994.

[War94b] M.O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the Conference on Visualization'94*, pages 326–333. IEEE Computer Society Press, 1994.

[Wat05] Martin Wattenberg. Baby Names Visualization, and Social Data Analysis. In *Proceedings of 2005 IEEE Symposium on Information Visualization (INFOVIS)*, pages 1–6, 2005.

[Wat09] R. J. C. Watt. Concordance 3.3, July 2009. `http://www.concordancesoftware.co.uk/`, Last Access Date: 2011-2-18.

[WB94] Pak Chung Wong and R. Daniel Bergeron. 30 Years of Multidimensional Multivariate Visualization. In *Scientific Visualization*, pages 3–33. IEEE Computer Society, 1994.

[WB96a] Pak Chung Wong and R. Daniel Bergeron. Multiresolution Multidimensional Wavelet Brushing. In *Proceedings of IEEE Visualization*, pages 141–148, 1996.

[WB96b] P.C. Wong and R.D. Bergeron. Multiresolution multidimensional wavelet brushing. In *Proceedings of the 7th conference on Visualization'96*, pages 141–ff. IEEE Computer Society Press, 1996.

[WB08] Martin Wattenberg and Fernanda B.Viegas. The Word Tree, an Interactive Visual Concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, 2008.

[Weg90a] Edward J. Wegman. Hyperdimensional Data Analysis Using Parallel Coordinates. *Journal of the American Statistical Association*, 85(411):664–672, 1990.

[Weg90b] E.J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990.

[WGK10] Matthew Ward, Georges Grinstein, and Daniel Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. A K Peters, Ltd., 2010.

[Wil96] Graham J. Wills. Selection: 524,288 Ways to Say "This is Interesting". In *Proceedings of the IEEE Symposium on Information Visualization*, pages 54–61. IEEE, 1996.

[WL97] Edward J. Wegman and Qiang Luo. High Dimensional Clustering Using Parallel Coordinates and the Grand Tour. *Computing Science and Statistics*, 28:352–360, 1997.

[WLWK08] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting TF-IDF Term Qeights As Making Relevance Decisions. *ACM Transactions on Information Systems*, 26(3):13:1–13:7, June 2008.

[Wor96] WordSmith.org. WordSmith Tools, 1996. `http://www.lexically.net/wordsmith/index.html`, Last Access Date: 2011-3-16.

[WPW+11] Yingcai Wu, Thomas Provan, Furu Wei, Shixia Liu, and Kwan-Liu Ma. Semantic-preserving word clouds by seam carving. *Comput. Graph. Forum*, 30(3):741–750, 2011.

[WTP+95] James A. Wise, James J. Thomas, Kelly Pennock, David Lantrip, Marc Pottier, Anne Schur, and Vern Crow. Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents. In *IEEE Symp. Information Visualization, InfoVis*, pages 51–58, 1995.

[XMD11] XMDV. Data Sets, 2011. `http://davis.wpi.edu/xmdv/index.html`.

[XW05] Rui Xu and D. Wunsch. Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks*, 16:645 – 678, 2005.

[YaKSJ07] Ji Soo Yi, Youn ah Kang, John T. Stasko, and Julie A. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Tranaction on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.

[Zai03] Feridun Zaimoglu. *William Shakespeare Othello* . Verlagshaus Monsenstein und Vannerdatp, 2003.

[Zil11] Brian Pytlik Zillig. TokenX: a text visualization, analysis, and play tool, 2011. `http://segonku.unl.edu/cocoon/tokenxcather/index.html?file=../xml/base.xml`, Last Access Date: 2011-2-18.

[ZK08] C. Ziemkiewicz and R. Kosara. The Shaping of Information by Visual Metaphors. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1269–1276, 2008.

[ZRL96] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An Efficient Data Clustering Method For Very Large Databases. In *Proceedings of ACM-SIGMOD International Conference of Management of Data*, pages 103–114, 1996.

[ZYQ+08] Hong Zhou, Xiaoru Yuan, Huamin Qu, Weiwei Cui, and Baoquan Chen. Visual Clustering in Parallel Coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.

[B10] Artur ilic and Bojana Dalbelo Basic. Visualization of Text Streams: A Survey . *Knowledge-Based and Intelligent Information and Engineering Systems*, 6277(6):31–43, 2010.